

ADAPTIVE WEAK APPROXIMATION OF DIFFUSIONS WITH JUMPS

E. MORDECKI[†], A. SZEPESSY[‡], R. TEMPONE[§] AND G. E. ZOURARIS[¶]

ABSTRACT. This work develops Monte Carlo Euler adaptive time stepping methods for the weak approximation problem of jump diffusion driven stochastic differential equations. The main result is the derivation of a new expansion for the computational error, with computable leading order term in a posteriori form, based on stochastic flows and discrete dual backward problems which extends the results in [34]. These expansions lead to efficient and accurate computation of error estimates. Adaptive algorithms for either stochastic time steps or quasi-deterministic time steps are described. Numerical examples show the performance of the proposed error approximation and of the described adaptive time-stepping methods.

February 2, 2008

1. INTRODUCTION

This work develops adaptive methods and proves a posteriori error expansions, with computable leading order term, for weak approximation of jump-diffusions driven stochastic differential equations.

1.1. Problem's setting. Consider $X = \{X(t) = (X^1(t), \dots, X^d(t)) : t \in [0, T]\}$, a d -dimensional stochastic process that is the solution of the stochastic differential equation

$$(1.1) \quad \begin{aligned} X(t) = X(0) &+ \int_0^t a(s, X(s^-)) ds + \sum_{\ell=1}^{\ell_0} \int_0^t b^\ell(s, X(s^-)) dW^\ell(s) \\ &+ \int_0^t \int_{\mathbf{Z}} c(s, X(s^-), z) p(ds, dz), \end{aligned}$$

on a time interval $[0, T]$ (see III.2c in [18], or, [17]). The randomness in the equation is generated by (i) an \mathcal{F}_0 -measurable d -dimensional random variable $X(0)$, (ii) a standard ℓ_0 -dimensional Wiener process $W = \{W(t) = (W^1(t), \dots, W^{\ell_0}(t)) : t \in [0, T]\}$, i.e. its coordinates are independent standard real valued Wiener processes, and (iii) a Poisson random measure p on $[0, T] \times \mathbf{Z}$, where $\mathbf{Z} \equiv \mathbb{R}^{\ell_1} \setminus \{0\}$, with deterministic time dependent intensity measure $q(dt, dz) = \lambda(t)dt \otimes \mu(t, dz)$. Here $\lambda : [0, T] \rightarrow [0, \lambda_{\max}]$ is the time intensity of jumps with $\lambda_{\max} \in (0, +\infty)$ and, for each $t \geq 0$ fixed, $\mu(t, dz)$ is a probability measure on \mathbf{Z} . All processes are defined on a stochastic basis $\mathcal{B} = (\Omega, \mathcal{F}_T, \{\mathcal{F}_t\}_{0 \leq t \leq T}, P)$, and, as usual in this framework, we assume that the Wiener process and the Poisson random measure are independent. The coefficients $a : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$, $b : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^{d \times \ell_0}$ and $c : [0, T] \times \mathbb{R}^d \times \mathbf{Z} \rightarrow \mathbb{R}^d$ are assumed to be Borel functions, satisfying regularity conditions defined in Lemma 2.1.

For a given scalar function $g : \mathbb{R}^d \rightarrow \mathbb{R}$, the goal of our work is to construct approximations to the expected value $E[g(X(T))]$ by a Monte Carlo Euler method (cf., e.g., [21], [26]). In what respects the

1991 *Mathematics Subject Classification.* 65C30, 65Y20, 65L50, 65H35.

Key words and phrases. Itô stochastic differential equations, diffusions with jumps, Monte Carlo Euler method, a posteriori error estimates, adaptive methods.

[†]Centro de Matemática, Facultad de Ciencias, Universidad de la República, Iguá 4225 C.P. 11400, Montevideo, Uruguay (mordecki@cmat.edu.uy).

[‡]Matematiska Institutionen, Kungl. Tekniska Högskolan, S-100 44 Stockholm, Sweden (szepeessy@nada.kth.se).

[§]Department of Mathematics and School of Computational Science and Information Technology, 400 Dirac Science Library, Florida State University, Tallahassee, FL 32306-4120 (rtempone@scs.fsu.edu).

[¶]Department of Mathematics, University of Crete, GR-714 09 Heraklion, Crete, Greece and Institute of Applied and Computational Mathematics, FO.R.T.H., GR-711 10 Heraklion, Crete, Greece (zouraris@math.uoc.gr).

dimension, although our developments are valid for all $d \in \mathbb{N}$, we have in mind relatively high values of d , taking into account the curse of dimensionality present in concurrent deterministic methods that solve partial integro-differential equations. The computational approach presented is common in computing option prices in mathematical finance and in simulating stochastic dynamics (cf., e.g., [8], [21], [22], [31]).

Remark 1.1 (Construction of the integral with respect to the Poisson random measure).

Consider a sequence $\mathbf{e}_1, \mathbf{e}_2, \dots$ of independent random variables with common exponential distribution with parameter 1. Define

$$(1.2) \quad \Lambda(t) = \int_0^t \lambda(s) ds, \quad t \in [0, T].$$

The number of jumps of the random Poisson measure $p(dt, dz)$ in an interval $[0, t]$ is determined as

$$\hat{N}(t) = \max \left\{ k : \sum_{j=1}^k \mathbf{e}_j \leq \Lambda(t) \right\},$$

and the total number of jumps in $[0, T]$ is denoted by $\hat{N} \equiv \hat{N}(T)$. The jump times of the Poisson measure can be defined by $\tau_0 = 0$, and $\tau_k = \Lambda^{-1}(\mathbf{e}_1 + \dots + \mathbf{e}_k)$ for $k = 1, \dots, \hat{N}$, and can be computed recursively by

$$(1.3) \quad \mathbf{e}_k = \int_{\tau_{k-1}}^{\tau_k} \lambda(s) dt, \quad \text{for } k = 1, \dots, \hat{N}.$$

Once the jump times are computed, we proceed to sample the marks $\{Z_k\}$, that, conditionally on the values of the jumps times, are independent random variables distributed respectively according to $\{\mu(\tau_k, dz)\}$. The random measure with intensity $q(dt, dz) = \lambda(t)dt \otimes \mu(t, dz)$ can then be constructed as

$$p(ds, dz) = \sum_{k=1}^{\hat{N}} \delta_{(\tau_k, Z_k)}(ds, dz),$$

and, consequently, the stochastic integral with respect to the Poisson random measure, i.e. the last term in (1.1), can be computed as

$$\int_0^t \int_{\mathbf{Z}} c(s, X(s^-), z) p(ds, dz) = \sum_{k=1}^{\hat{N}(t)} c(\tau_k, X(\tau_k^-), Z_k), \quad t \in [0, T].$$

Remark 1.2. Since the time intensity $\lambda(t)$ is deterministic, the jump times $\{\tau_k\}$ can be directly obtained before solving for the process X ; it is enough then to find the function Λ by accurately integrating (1.2) and then successively finding $\{\tau_k\}$ by solving (1.3). Furthermore, if λ is just a constant then we simply sample $\tau_k = \mathbf{e}_1 + \dots + \mathbf{e}_k$ from a sequence $\{\mathbf{e}_k\}$ of independent random variables with common exponential distribution with parameter λ .

1.2. The Monte Carlo Euler method. We now present the Monte Carlo Euler time stepping algorithm which will be the building block for adaptive algorithms, with either quasi-deterministic or stochastic time steps. For each realization we first construct the jumps and its marks, and conditioned on this information we construct the approximate solution \bar{X} as follows.

Monte Carlo Euler time stepping algorithm

Input: Give a number $N + 1$ of time nodes $0 = \tilde{t}_0 < \tilde{t}_1 < \dots < \tilde{t}_N = T$, and sample jump times $0 < \tau_1(\omega) < \dots < \tau_{\hat{N}(\omega)}(\omega) < T$ with corresponding marks $Z_1(\omega), \dots, Z_{\hat{N}(\omega)}(\omega)$, as explained in Remark 1.1.

Set the jump counter $k = 1$.

Time stepping: Consider an augmented partition given by the union

$$\{t_n(\omega)\}_{n=0}^{N_A(\omega)} = \{\tilde{t}_n\}_{n=0}^N \cup \{\tau_n(\omega)\}_{n=0}^{\hat{N}(\omega)}$$

where $N_A(\omega) = N + \hat{N}(\omega)$ (*a.s.*) is the number of time steps.

Sample $X_0(\omega)$ and set the initial condition $\bar{X}(t_0, \omega) = X_0(\omega)$.

For time steps $n = 0, \dots, N_A(\omega) - 1$

 Compute the remainder approximate grid values, $\bar{X}(t_{n+1}, \omega)$,
 by first constructing the left limit value of the approximated process,

$$(1.4) \quad \begin{aligned} \bar{X}(t_{n+1}^-, \omega) &= \bar{X}(t_n, \omega) + a(t_n, \bar{X}(t_n, \omega))(t_{n+1} - t_n) \\ &\quad + \sum_{\ell=1}^{\ell_0} b^\ell(t_n, \bar{X}(t_n, \omega))(W^\ell(t_{n+1}, \omega) - W^\ell(t_n, \omega)). \end{aligned}$$

 When needed, introduce the correction due to jump discontinuities:

if $(t_{n+1} = \tau_k(\omega))$ **then**

$$(1.5) \quad \begin{aligned} \bar{X}(t_{n+1}, \omega) &= \bar{X}(t_{n+1}^-, \omega) + c(t_{n+1}, \bar{X}(t_{n+1}^-, \omega), Z_k(\omega)), \\ &\text{increase } k \text{ to } k + 1 \end{aligned}$$

else

$$(1.6) \quad \bar{X}(t_{n+1}, \omega) = \bar{X}(t_{n+1}^-, \omega),$$

end-if

end-for

When the initial time nodes $\{\tilde{t}_0, \dots, \tilde{t}_N\}$ are the same for all realizations, we refer to *quasi-deterministic* time steps, or sometimes, simply *deterministic* time steps. Otherwise, we speak about *stochastic* time steps.

Besides, in the particular case when $a = b = 0$ the approximate process \bar{X} has the same law as X , due to the form of the grid proposed in the numerical method (see [15]). A similar situation occurs when only $b = 0$: there, conditioned to the realizations of \hat{N} , a higher order method for ODE integration should be used to approximate \bar{X} between jumps. It must be also noticed that the jump intensity λ does not depend on the current value $X(t)$ of the process. Such a dependence carries the necessity of implementing a different Monte Carlo Euler algorithm, where the jumps and its marks are sampled simultaneously with the trajectory of the process, generating an additional error in the approximation scheme, as it does not seems possible to have an exact simulation of the jump structure in law, something that is possible in the present case.

1.3. Error Control and Adaptivity. The aim, for a given $\text{TOL} > 0$, is to choose the size of time steps, either quasi-deterministic or stochastic,

$$\Delta \tilde{t}_n = \tilde{t}_{n+1} - \tilde{t}_n, \quad n = 0, \dots, N - 1,$$

and the number M of independent identically distributed samples $\{\bar{X}(\cdot, \omega_j)\}_{j=1}^M$ such that the computational work, defined as M times the average of timesteps, i.e. $M \times E[N_A] = M \times (E[N] + \Lambda(T))$, is

minimal, constrained by the condition that the computational error \mathcal{E}_C defined by

$$(1.7) \quad \mathcal{E}_C = E[g(X(T))] - \frac{1}{M} \sum_{j=1}^M g(\bar{X}(T, \omega_j)),$$

is such that the event

$$(1.8) \quad |\mathcal{E}_C| \leq \text{TOL}$$

has a probability close to one. The computational error \mathcal{E}_C naturally separates as the sum of the (deterministic) time discretization error \mathcal{E}_T and the statistical error \mathcal{E}_S given by

$$(1.9) \quad \mathcal{E}_T = E[g(X(T))] - E[g(\bar{X}(T))],$$

$$(1.10) \quad \mathcal{E}_S = E[g(\bar{X}(T))] - \frac{1}{M} \sum_{j=1}^M g(\bar{X}(T, \omega_j)).$$

The time steps to construct the trajectories \bar{X} are determined from statistical approximations of the time discretization error \mathcal{E}_T ; while the number M of independent realizations of \bar{X} , is determined from the statistical error \mathcal{E}_S . Therefore, the number M of realizations can be asymptotically determined by standard limit theorems for sums of independent random variables.

Efficient adaptive time stepping methods, with theoretical basis, use a posteriori error information, since the a priori knowledge can usually not be as precise as the a posteriori. The present work develops adaptive time stepping methods by proving in Theorem 2.3 and Theorem 3.3 error estimates of \mathcal{E}_T with leading order terms in computable a posteriori form.

The main reference of Theorems 2.3 and 3.3 is the work by Szepessy, Tempone and Zouraris [34] where similar techniques were applied considering the solution of a stochastic differential equations, i.e. our case when $c = 0$. The general inspiration of the presented results is the work by Talay and Tubaro in [39] and its subsequent extension by Protter and Talay in [38] to stochastic differential equations driven by Lévy processes.

The main new idea here is the extension of the efficient use of stochastic flows and dual functions to obtain the error expansion with computable leading order term in Theorems 2.3 and 3.3 in presence of jumps. The use of dual functions is standard in optimal control theory and in particular for adaptive mesh control for ordinary and partial differential equations, see [3], [19], [20], [12], and [6], and was successfully applied in the probabilistic context in [34], [29], [11] and in [27].

Concerning time steps, this work proposes adaptive algorithms that use either

- “quasi-deterministic” time steps (see section 4.3), in the sense that, although the grid is random, its randomness only depends on the jumps, being in this sense a strict generalization of the deterministic time step algorithm in [34].
- “stochastic” time steps, meaning that the grid is random also with respect to the Wiener measure, similarly as with the stochastic time steps introduced in [34].

Theorem 2.3 describes the basis of an adaptive algorithm to estimate the computational error; the deterministic time steps are then chosen by

$$(\Delta \tilde{t}_n)^2 |E[\rho(\tilde{t}_n, \omega)]| = \text{constant},$$

– observe that we can only control the deterministic points \tilde{t}_n and not the whole augmented grid $\{t_n(\omega)\}_{n=0}^{N_A(\omega)}$ – where $\rho(\tilde{t}_n, \omega)$ is the function defined in (2.9), based on the weight functions φ and φ' defined in (A.1) – (A.4). Provided the path $\bar{X}(t_n)$, $n = 0, \dots, N_A$, is stored, the leading order error bound can be evaluated by solving, step by step, the two backward problems (A.1) – (A.4). The backward evolutions (A.1) – (A.4) of the weight functions φ and φ' avoid solving for the two variables t, s present in $\frac{\partial \bar{X}(t)}{\partial x(s)}$, which appears in the forward t –evolution equation for $\frac{\partial \bar{X}(t)}{\partial x(s)}$ in the identity $\varphi_i(t_n) = \partial_j g(\bar{X}(T)) \partial \bar{X}_j(t_n) / \partial x_i(t_n)$, cf. (2.34). A solution with two variables s and t would require work of the order N^2 for each realization, instead of the corresponding work of the order N in Theorem 2.3.

The second algorithm, presented in Section 4.4, is based on the expansion derived in Theorem 3.3 and uses time steps which may vary for different realizations of the Wiener process. Stochastic time steps are advantageous for problems with singularities at random times. The idea in this case is to choose the steps by

$$(\Delta \tilde{t}_n)^2 |\rho(\tilde{t}_n, \omega)| = \text{constant},$$

(where, in comparison with the previous algorithm, there is no expectation), and this is achieved through a test performed at each interval of each realization, to decide whether to refine or not the given interval. In this case, when a node is added, the interpolation is carried through the consideration of a Brownian bridge to obtain the value of the approximated process \bar{X} in the new added point.

Besides, since their use entails more work per realization than the deterministic time steps they should be judiciously used. A natural application of stochastic time steps appears in the weak approximation of killed diffusions, see [11]. The optimal stochastic steps depend on the whole solution $\bar{X}(t)$, $0 < t < T$, and in particular the step $\Delta t(t)$ at time t depends also on $W(u)$ for $t < u$. In stochastic analysis the concept of *adaptedness* means, intuitively, that the values of the process at time t depend only on the events generated by the sources of randomness up to time t , i.e. do not depend on future events. In numerical analysis a method is said to be *adaptive* when the approximate solution is used to control the error, e.g. to determine the time steps. Our stochastic time stepping algorithm is in this sense adaptive and non adapted, since the time steps $\Delta t(t)$ depends on values of $W(u)$ for $t < u$, i.e. on future values of the Wiener process. The stochastic time stepping algorithm (that achieves higher precision) requires, on one side, an additional theoretical developement including the introduction of Malliavin derivatives, and, on the other, requires the approximate computation of derivatives of second order. This second requirement is performed introducing the second derivatives φ'' of the fluxes (the procedure is described in the Appendix). This computation is performed also in linear time.

The focus in the paper is on error estimates for weak convergence of stochastic differential equations for diffusions with jumps. Two particular important cases must be distinguished. In first place, taking $c = 0$ we obtain previous estimates in [34]. More relevant in this instance is the situation when $a = b = 0$ and $c \neq 0$. In this case, we are considering a pure jump process, with finite number of jumps in the time interval $[0, T]$. From the stochastic point of view, the situation is simpler, and the process \bar{X} has exactly the same distribution as X . This means that the time discretization error is null, and the problem reduces to controlling the statistical error.

Furthermore, the deterministic problems associated with the computation of the expected value $E[g(X(T))]$ via the Feynman-Kac formula are non local, in the sense that they involve integro-differential equations. Direct discretization of such time dependent integro-differential equations needs to approximate integrals at each time step, while the Euler Monte Carlo method avoids these expensive computations.

When d is small and the Fourier transform of $X(t)$ is known (for instance, X is a Lévy process), it is efficient to approximate $E[g(X(T))]$ based on Parseval's identity for this transform and the Fourier transform of g [23, 9]. The work [2] uses operator splitting to approximate the integral term explicitly in Fourier space, while approximating the other terms in the equation implicitly. The work [25] discretizes the partial integro-differential equation by the θ -scheme in time and a wavelet Galerkin method in space. The resulting full Galerkin matrix is then replaced with a sparse matrix in the wavelet basis, and the linear systems for each time step are solved approximatively with GMRES in linear complexity. The deterministic algorithm gives optimal convergence rates, up to logarithmic terms, for the computed solution in the same complexity as finite difference approximations of the standard Black-Scholes equation. Other works include [24], where weak convergence schemes are analyzed, [16] where two implicit numerical methods for diffusion with jumps are presented, analyzing strong convergence and nonlinear stability; and [7] that contains a survey, including some new results on weak approximation for jump diffusion equations.

The technique used here is based on the Kolmogorov's backward equation developed in [36] and [37] to analyze uniqueness and dependence on initial conditions for weak solutions of stochastic differential equations with jumps. The rest of the paper is as follows. Section 2 proves error estimates for quasi-deterministic time steps. Section 3 proves error estimates for stochastic time steps. Section 4 presents

implementations of adaptive algorithms and, finally, Section 5 includes results from numerical experiments.

2. AN ERROR ESTIMATE OF THE COMPUTATIONAL ERROR WITH DETERMINISTIC TIME STEPS

In this section we present in Theorem 2.3 an error expansion in a posteriori computable form for the time discretization error \mathcal{E}_T . The starting point for the analysis is Lemma 2.1 below. It uses the fact that the Euler method can be extended, for theoretical purposes only, by

$$(2.1) \quad \overline{X}(t) - \overline{X}(t_n) = \int_{t_n}^t \overline{a}(s; \overline{X}) ds + \sum_{\ell=1}^{\ell_0} \int_{t_n}^t \overline{b}^\ell(s; \overline{X}) dW^\ell(s) \quad \forall t \in [t_n, t_{n+1}),$$

where \overline{a} and \overline{b}^ℓ are the piecewise constant stochastic approximations

$$(2.2) \quad \overline{a}(s; \overline{X}) = a(t_n, \overline{X}(t_n)) \quad \text{and} \quad \overline{b}^\ell(s; \overline{X}) = b^\ell(t_n, \overline{X}(t_n)) \quad \text{for } s \in [t_n, t_{n+1}).$$

Observe that the presence of jumps is realized in \overline{X} in (1.5), making possible not to introduce a modified coefficient for c . For simplicity we introduce the notation

$$\partial_k \equiv \frac{\partial}{\partial x_k}, \quad \partial_{ki} \equiv \frac{\partial^2}{\partial x_k \partial x_i}, \quad \dots \quad \partial_t \equiv \frac{\partial}{\partial t},$$

and use the summation convention, i.e., if the same subscript appears twice in a term, the term denotes the sum over the range of this subscript, e.g. $c_{ik} \partial_k b_j \equiv \sum_{k=1}^d c_{ik} \partial_k b_j$, and consequently

$$d_{ij} \equiv \frac{1}{2} \sum_{\ell=1}^{\ell_0} b_i^\ell b_j^\ell = \frac{1}{2} b_i^\ell b_j^\ell, \quad \overline{d}_{ij} \equiv \frac{1}{2} \sum_{\ell=1}^{\ell_0} \overline{b}_i^\ell \overline{b}_j^\ell = \frac{1}{2} \overline{b}_i^\ell \overline{b}_j^\ell.$$

For a derivative ∂_α the notation $|\alpha|$ is its order.

Lemma 2.1. *Suppose that for some $m_0 > [d/2] + 10$ there are positive constants k and C such that*

- (i) $g \in \mathcal{C}_{loc}^{m_0}(\mathbb{R}^d)$ with $|\partial_\alpha g(x)| \leq C(1 + |x|^k)$ for all $|\alpha| \leq m_0$,
- (ii) $E[|X(0)|^{2k+d+1} + |\overline{X}(0)|^{2k+d+1}] \leq C$, and
- (iii) a and b are bounded in $\mathcal{C}^{m_0}([0, T] \times \mathbb{R}^d)$ and the same holds uniformly for $c(\cdot, \cdot, z)$ for all $z \in \mathbf{Z}$.
- (iv) the initial data $X(0)$ and its approximation $\overline{X}(0)$ have the same distribution.

Then, the time discretization error in (1.9) satisfy

$$(2.3) \quad \mathcal{E}_T = \int_0^T E[(a_k(t, \overline{X}(t^-)) - \overline{a}_k(t; \overline{X})) \partial_k u(t, \overline{X}(t^-)) + (d_{ij}(t, \overline{X}(t^-)) - \overline{d}_{ij}(t; \overline{X})) \partial_{ij} u(t, \overline{X}(t^-))] dt$$

where

$$(2.4) \quad u(t, x) = E[g(X(T)) \mid X(t) = x].$$

is the cost to go function.

Remark 2.2. *We can relax condition (iv) in the assumptions of the Lemma, and get an additional term of the form $E[u(0, X(0)) - u(0, \overline{X}(0))]$ in the error expansion in (2.3).*

Proof. There exists a unique solution $u \in \mathcal{C}_{loc}^{1,6}([0, T] \times \mathbb{R}^d)$ of the Kolmogorov backward equation

$$(2.5) \quad \begin{aligned} \mathcal{L}^X u(t, x) &\equiv \partial_t u(t, x) + a_k \partial_k u(t, x) + d_{kn} \partial_{kn} u(t, x) + \lambda(t) \int_{\mathbf{Z}} [u(t, x + c(t, x, z)) - u(t, x)] \mu(t, dz) = 0 \\ u(T, \cdot) &= g, \end{aligned}$$

satisfying the polynomial growth condition

$$\max_{0 \leq t \leq T} |\partial_\alpha u(t, x)| \leq C \left(1 + |x|^{k + \frac{d+1}{2}} \right) \quad \forall |\alpha| \leq 6,$$

for some positive k and C (cf. [14]). The Feynman-Kac formula without potential, implies that the solution u of (2.5) can be represented by the expected value in (2.4). The Itô formula applied to (2.1) (cf. [17], p. 66) gives

$$\begin{aligned} u(T, \overline{X}(T)) - u(0, \overline{X}(0)) &= \int_0^T \{ \partial_t u(t, \overline{X}(t^-)) + \overline{a}_i(t; \overline{X}) \partial_i u(t, \overline{X}(t^-)) + \overline{d}_{ij} \partial_{ij} u(t, \overline{X}(t^-)) \\ &\quad + \lambda(t) \int_{\mathbf{Z}} [u(t, \overline{X}(t^-) + c(t, \overline{X}(t^-), z)) - u(t, \overline{X}(t^-))] \mu(t, dz) \} dt \\ &\quad + \int_0^T \overline{b}_i^\ell(t; \overline{X}) \partial_i u(t, \overline{X}(t^-)) dW^\ell(t) \\ &\quad + \int_0^T \int_{\mathbf{Z}} [u(t, \overline{X}(t^-) + c(t, \overline{X}(t^-), z)) - u(t, \overline{X}(t^-))] (p(dt, dz) - q(dt, dz)) \end{aligned}$$

which, combined with (2.5) to substitute $\partial_t u(t, \overline{X}(t^-))$, yields

$$\begin{aligned} (2.6) \quad u(0, \overline{X}(0)) - g(\overline{X}(T)) &= \int_0^T (a_i(t, \overline{X}(t^-)) - \overline{a}_i(t; \overline{X})) \partial_i u(t, \overline{X}(t^-)) dt \\ &\quad + \int_0^T (d_{ij}(t, \overline{X}(t^-)) - \overline{d}_{ij}(t; \overline{X})) \partial_{ij} u(t, \overline{X}(t^-)) dt \\ &\quad - \int_0^T \overline{b}_i^\ell(t; \overline{X}) \partial_i u(t, \overline{X}(t^-)) dW^\ell(t) \\ &\quad - \int_0^T \int_{\mathbf{Z}} [u(t, \overline{X}(t^-) + c(t, \overline{X}(t^-), z)) - u(t, \overline{X}(t^-))] (p(dt, dz) - q(dt, dz)). \end{aligned}$$

The expected value of the last two integrals is zero, the first one by the martingale property of Itô integrals, and the second one due to the fact that $p(dt, dz) - q(dt, dz)$ is a compensated random measure (i.e. a martingale measure). Condition (iv) in the Lemma, and the representation of u in (2.4) show that

$$E[u(0, \overline{X}(0))] = E[u(0, X(0))] = E[g(X(T))].$$

Therefore, taking expected values in both sides of (2.6) we arrive at the error representation (2.3). \square

Lemma 2.1 is combined with stochastic flows to derive the a posteriori error expansion in Theorem 2.3 below. This error expansion is based on the variations of the processes X and \overline{X} . For a process \overline{X} , the first variation of a function $F(\overline{X}(T))$ with respect to a perturbation in the initial location of the path \overline{X} , at time s , is denoted by

$$(2.7) \quad F'(T; s) = \partial_{x(s)} F(\overline{X}(T)) \equiv (\partial_1 F(\overline{X}(T); \overline{X}(s) = x), \dots, \partial_d F(\overline{X}(T); \overline{X}(s) = x)).$$

The proof of Theorem 2.3 uses mainly that the error in replacing $g(X(T))$ in Lemma 2.1 by $g(\overline{X}(T))$, in the representation (2.4) of $\partial_\alpha u$, yields the small deterministic remainder term $\int_0^T O((\Delta t)^2) dt$ in (2.8) of Theorem 2.3, which is analogous to the $\mathcal{O}(N^{-2})$ term in Talay and Tubaro's expansion, cf. [39], and needs some a priori estimate to be controlled. Lemma 2.1 can be applied to estimate this error. The second important ingredient in the proof is the Markov property of \overline{X} satisfied at the discrete times t_n . Based on the fact that $\overline{X}(t_n)$ is \mathcal{F}_{t_n} measurable, the nested expected values

$$E[a_j(t_n, \overline{X}(t_n)) \partial_{x_j(t_n)} E[g(\overline{X}(T)) \mid \mathcal{F}_{t_n}]]$$

in (2.3) can, by the definition of φ and its implication (2.34), be decoupled to

$$E[a_j(t_n, \overline{X}(t_n)) \varphi_j(t_n)],$$

which reduces the computational complexity substantially, see Lemma 2.8.

Theorem 2.3 (Error expansion with deterministic time steps). *Suppose that a , b , g , X and \overline{X} , satisfy the assumptions in Lemma 2.1. Then, the time discretization error in (1.9) has the expansion*

$$(2.8) \quad \mathcal{E}_T = E \left[\sum_{m=0}^{N-1} \rho(\tilde{t}_m, \omega) (\Delta \tilde{t}_m)^2 \right] + E \left[\sum_{m=0}^{N-1} (\Delta \tilde{t}_m)^2 \left\{ \mathcal{O}(\Delta \tilde{t}_m) + \sum_{m=n}^{N-1} \mathcal{O}((\Delta \tilde{t}_m)^2) \right\} \right]$$

where the leading order error term is in computable a posteriori form.

$$(2.9) \quad \rho(\tilde{t}_m, \omega) \equiv \frac{1}{2} \sum_{n \in \mathcal{J}_m} [(a_i(t_{n+1}, \overline{X}(t_{n+1}^-), \omega) - a_i(t_n, \overline{X}(t_n, \omega))) \varphi_i(t_{n+1}^-, \omega)] \frac{\Delta t_n}{(\Delta \tilde{t}_m)^2} \\ + \frac{1}{2} \sum_{n \in \mathcal{J}_m} [(d_{ik}(t_{n+1}, \overline{X}(t_{n+1}^-), \omega) - d_{ik}(t_n, \overline{X}(t_n, \omega))) \varphi'_{ik}(t_{n+1}^-, \omega)] \frac{\Delta t_n}{(\Delta \tilde{t}_m)^2}$$

with $\mathcal{J}_m \equiv \{n : \tilde{t}_m \leq t_n < \tilde{t}_{m+1}\}$, $m = 0, \dots, N-1$, and based on the discrete dual functions $\varphi(t_n) \in \mathbb{R}^d$ and $\varphi'(t_n) \in \mathbb{R}^{d \times d}$, which are determined as follows. The function φ and its first variation

$$(2.10) \quad \varphi'_{ik}(t_n, \omega) = \partial_{x_k(t_n)} \varphi_i(t_n, \omega) \equiv \frac{\partial \varphi_i(t_n; \overline{X}(t_n) = x)}{\partial x_k}$$

satisfies (A.1) – (A.4).

Remark 2.4. When implementing an algorithm based on this result the expectation of the sum of errors in (2.8) is approximated by the mean of the errors along the M simulated trajectories, i.e.:

$$E \left[\sum_{m=0}^{N-1} \rho(\tilde{t}_m, \omega) (\Delta \tilde{t}_m)^2 \right] \sim \frac{1}{M} \sum_{j=1}^M \sum_{m=0}^{N-1} \frac{\rho(\tilde{t}_m, \omega_j)}{M} (\Delta \tilde{t}_m)^2.$$

The statistical error of this approximation can be expressed as

$$E \left[\sum_{m=0}^{N-1} \rho(\tilde{t}_m, \omega) (\Delta \tilde{t}_m)^2 \right] - \frac{1}{M} \sum_{j=1}^M \sum_{m=0}^{N-1} \frac{\rho(\tilde{t}_m, \omega_j)}{M} (\Delta \tilde{t}_m)^2 = \int_0^T (I_M + II_M) dt$$

where the distributions of the statistical errors $\sqrt{M}I_M$ and $\sqrt{M}II_M$ weakly converge to normal distributions with mean zero and time interval dependent variances given by

$$\text{Var} \left[\sum_{n \in \mathcal{J}_m} (a_i(t_{n+1}, \overline{X}(t_{n+1}^-)) - a_i(t_n, \overline{X}(t_n))) \varphi_i(t_{n+1}^-) \right] = \mathcal{O}(\Delta \tilde{t}_m),$$

and

$$\text{Var} \left[\sum_{n \in \mathcal{J}_m} (d_{ik}(t_{n+1}, \overline{X}(t_{n+1}^-)) - d_{ik}(t_n, \overline{X}(t_n))) \varphi'_{ik}(t_{n+1}^-) \right] = \mathcal{O}(\Delta \tilde{t}_m),$$

respectively.

Proof of Theorem 2.3. The main content of Theorem 2.3 is the replacement of the (non computable) estimate in Lemma 2.1 by an expansion with computable leading order term. For this purpose the derivatives of the expected value

$$\partial_\alpha u(x, t) = \partial_\alpha E[g(X(T)) \mid X(t) = x]$$

appearing in the integral in Lemma 2.1 are approximated by the corresponding derivatives of

$$\bar{u}(x, t) \equiv E[g(\overline{X}(T)) \mid \overline{X}(t) = x],$$

that depends on the simulated solution \overline{X} . The proof is divided into three steps:

- (i) in Lemma 2.5 we estimate the quadrature error;
- (ii) in Lemma 2.6 we bound the error in replacing $\partial_\alpha u$ by $\partial_\alpha \bar{u}$ with the use of stochastic flows and its variations;
- (iii) in Lemma 2.8 we use the discrete dual functions φ and φ' (that solve the backward evolution problems see (A.1) – (A.4) in the Appendix) to derive a computable representation of $\partial_\alpha \bar{u}$.

We begin with the first step.

Lemma 2.5 (Quadrature approximation). *Suppose that the assumptions in Lemma 2.1 hold. Let $\mathcal{J}_m \equiv \{n : \tilde{t}_m \leq t_n < \tilde{t}_{m+1}\}$, $m = 0, \dots, N-1$. Then the quadrature error terms satisfy*

$$\begin{aligned} & \int_{\tilde{t}_m}^{\tilde{t}_{m+1}} E[(a_i(t, \bar{X}(t^-)) - \bar{a}_i(t; \bar{X})) \partial_i u(t, \bar{X}(t^-))] dt \\ & - E\left[\sum_{n \in \mathcal{J}_m} (a_i(t_{n+1}, \bar{X}(t_{n+1}^-)) - a_i(t_n, \bar{X}(t_n))) \partial_i u(t_{n+1}, \bar{X}(t_{n+1}^-)) \frac{\Delta t_n}{2}\right] = O((\Delta \tilde{t}_m)^3), \end{aligned}$$

and

$$\begin{aligned} & \int_{\tilde{t}_m}^{\tilde{t}_{m+1}} E[(d_{ij}(t, \bar{X}(t^-)) - \bar{d}_{ij}(t; \bar{X})) \partial_{ij} u(t, \bar{X}(t^-))] dt \\ & - E\left[\sum_{n \in \mathcal{J}_m} (d_{ij}(t_{n+1}, \bar{X}(t_{n+1}^-)) - d_{ij}(t_n, \bar{X}(t_n))) \partial_{ij} u(t_{n+1}, \bar{X}(t_{n+1}^-)) \frac{\Delta t_n}{2}\right] = O((\Delta \tilde{t}_m)^3). \end{aligned}$$

Proof. Denote by \mathcal{G} the σ -algebra generated by the jumps and marks in $[0, T]$ constructed in Remark 1.1. Then

$$\begin{aligned} (2.11) \quad & \int_{\tilde{t}_m}^{\tilde{t}_{m+1}} E[(d_{ij}(t, \bar{X}(t^-)) - \bar{d}_{ij}(t; \bar{X})) \partial_{ij} u(t, \bar{X}(t^-))] dt \\ & = E\left[\sum_{n \in \mathcal{J}_m} \int_{t_n}^{t_{n+1}} E[(d_{ij}(t, \bar{X}(t^-)) - \bar{d}_{ij}(t; \bar{X})) \partial_{ij} u(t, \bar{X}(t^-)) \mid \mathcal{G}] dt\right]. \end{aligned}$$

Observe that in $[t_n, t_{n+1})$ the conditioned process \bar{X} has no jump discontinuities, and introduce the notations

$$\begin{aligned} h(t, \bar{X}(t^-)) & \equiv (d_{ij}(t, \bar{X}(t^-)) - \bar{d}_{ij}(t; \bar{X})) \partial_{ij} u(t, \bar{X}(t^-)), \\ \bar{h}(t) & \equiv \frac{t - t_n}{\Delta t_n} (d_{ij}(t_{n+1}, \bar{X}(t_{n+1}^-)) - \bar{d}_{ij}(t_n; \bar{X})) \partial_{ij} u(t_{n+1}, \bar{X}(t_{n+1}^-)). \end{aligned}$$

Then the quadrature error satisfies

$$\begin{aligned} & \int_{t_n}^{t_{n+1}} E[h(t, \bar{X}(t^-)) - \bar{h}(t) \mid \mathcal{G}] dt = \int_{t_n}^{t_{n+1}} E[(d_{ij}(t, \bar{X}(t^-)) - \bar{d}_{ij}(t; \bar{X})) \partial_{ij} u(t, \bar{X}(t^-)) \mid \mathcal{G}] dt \\ & - E[(d_{ij}(t_{n+1}, \bar{X}(t_{n+1}^-)) - \bar{d}_{ij}(t_n; \bar{X})) \partial_{ij} u(t_{n+1}, \bar{X}(t_{n+1}^-)) \mid \mathcal{G}] \frac{\Delta t_n}{2}, \end{aligned}$$

and $E[\bar{h}(t) \mid \mathcal{G}]$ is the linear nodal projection of the smooth function $E[h(t, \bar{X}(t^-)) \mid \mathcal{G}]$ in the interval $[t_n, t_{n+1})$. Therefore, a standard interpolation estimate yields

$$(2.12) \quad \left| \int_{t_n}^{t_{n+1}} E[h(t, \bar{X}(t^-)) - \bar{h}(t) \mid \mathcal{G}] dt \right| \leq \frac{1}{8} (\Delta t_n)^2 \int_{t_n}^{t_{n+1}} \left| \frac{d^2}{dt^2} E[h(t, \bar{X}(t^-)) \mid \mathcal{G}] \right| dt.$$

Denoting $\mathcal{L}h \equiv \partial_t h + \bar{a}_i \partial_i h + \bar{d}_{ij} \partial_{ij} h$, Itô's formula and condition (iii) in Lemma 2.1 show that there exist constants C_1, C_2 such that, for $t \in (t_n, t_{n+1})$

$$\begin{aligned} (2.13) \quad & \frac{d}{dt} E[h(t, \bar{X}(t^-)) \mid \mathcal{G}] = E[\mathcal{L}h(t, \bar{X}(t^-)) \mid \mathcal{G}] \leq C_1, \\ & \frac{d^2}{dt^2} E[h(t, \bar{X}(t^-)) \mid \mathcal{G}] = E[\mathcal{L}^2 h(t, \bar{X}(t^-)) \mid \mathcal{G}] \leq C_2, \end{aligned}$$

which combined with (2.11) and (2.12) proves the estimate of the diffusion term in the lemma. The estimate of the drift term follows analogously. \square

In the second step of the proof the derivative $\partial_\alpha u$ and its approximation $\partial_\alpha \bar{u}$ are evaluated respectively through expected values of *stochastic flows* of X and \bar{X} and its *variations*, that we now introduce. Recall

the definition (2.7) of the first variation, and let

$$\delta_{ik} \equiv \begin{cases} 0 & i \neq k, \\ 1 & i = k. \end{cases}$$

The following equation for the first variation of the process X at time $s > t$ hold:

$$(2.14) \quad \begin{aligned} dX'_{ij}(s) &= \partial_k a_i(s, X(s^-)) X'_{kj}(s^-) ds + \partial_k b_i^\ell(s, X(s^-)) X'_{kj}(s^-) dW^\ell(s) \\ &\quad + \int_{\mathbf{Z}} \partial_k c_i(s, X(s^-), z) X'_{kj}(s^-) p(ds, dz), \\ X'_{ij}(t) &= \delta_{ij}. \end{aligned}$$

Similarly, for the second variation of the process X at time $s > t$ we have:

$$(2.15) \quad \begin{aligned} dX''_{ijn}(s) &= [\partial_k a_i(s, X(s^-)) X''_{kjn}(s^-) + \partial_{kr} a_i(s, X(s^-)) X'_{kj}(s^-) X'_{rn}(s^-)] ds \\ &\quad + [\partial_k b_i^\ell(s, X(s^-)) X''_{kjn}(s^-) + \partial_{kr} b_i^\ell(s, X(s^-)) X'_{kj}(s^-) X'_{rn}(s^-)] dW^\ell(s) \\ &\quad + \int_{\mathbf{Z}} [\partial_k c_i(s, X(s^-), z) X''_{kjn}(s^-) + \partial_{kr} c_i(s, X(s^-), z) X'_{kj}(s^-) X'_{rn}(s^-)] p(ds, dz), \\ X''_{ijn}(t) &= 0. \end{aligned}$$

For the third variation of the process X at time $s > t$ we have:

$$(2.16) \quad \begin{aligned} dX'''_{ijnm}(s) &= [\partial_k a_i(s, X(s^-)) X'''_{kijnm}(s^-) + \partial_{kr} a_i(s, X(s^-)) X'_{kj}(s^-) X''_{rnm}(s^-) \\ &\quad + \partial_{kr} a_i(s, X(s^-)) X'_{kn}(s^-) X''_{rjm}(s^-) + \partial_{kr} a_i(s, X(s^-)) X'_{km}(s^-) X''_{rjn}(s^-) \\ &\quad + \partial_{krv} a_i(s, X(s^-)) X'_{kj}(s^-) X'_{rn}(s^-) X'_{vm}(s^-)] ds \\ &\quad + [\partial_k b_i^\ell(s, X(s^-)) X'''_{kijnm}(s^-) + \partial_{kr} b_i^\ell(s, X(s^-)) X'_{kj}(s^-) X''_{rnm}(s^-) \\ &\quad + \partial_{kr} b_i^\ell(s, X(s^-)) X'_{kn}(s^-) X''_{rjm}(s^-) + \partial_{kr} b_i^\ell(s, X(s^-)) X'_{km}(s^-) X''_{rjn}(s^-) \\ &\quad + \partial_{krv} b_i^\ell(s, X(s^-)) X'_{kj}(s^-) X'_{rn}(s^-) X'_{vm}(s^-)] dW^\ell(s) \\ &\quad + \int_{\mathbf{Z}} [\partial_k c_i(s, X(s^-)) X'''_{kijnm}(s^-) + \partial_{kr} c_i(s, X(s^-), z) X'_{kj}(s^-) X''_{rnm}(s^-) \\ &\quad + \partial_{kr} c_i(s, X(s^-), z) X'_{kn}(s^-) X''_{rjm}(s^-) + \partial_{kr} c_i(s, X(s^-), z) X'_{km}(s^-) X''_{rjn}(s^-) \\ &\quad + \partial_{krv} c_i(s, X(s^-), z) X'_{kj}(s^-) X'_{rn}(s^-) X'_{vm}(s^-)] p(ds, dz), \\ X'''_{ijnm}(t) &= 0, \end{aligned}$$

and similarly for the fourth variation of the process X at time $s > t$:

$$(2.17) \quad dX''''_{ijnmp} = \dots, \quad X''''_{ijnmp}(t) = 0.$$

This equations imply the representation of the derivatives of expectations with stochastic flows as follows (cf. [33] and [35]). For the first derivatives:

$$(2.18) \quad \partial_k u(t, x) = E[\partial_i g(X(T)) X'_{ik}(T) \mid X'_{ij}(t) = \delta_{ij}, X(t) = x],$$

for the second derivatives:

$$(2.19) \quad \partial_{kn} u(t, x) = E[\partial_i g(X(T)) X''_{ikn}(T) + \partial_{ir} g(X(T)) X'_{ik}(T) X'_{rn}(T) \mid X''_{ikn}(t) = 0, X'_{ij}(t) = \delta_{ij}, X(t) = x],$$

for the third derivatives:

$$(2.20) \quad \begin{aligned} \partial_{knm} u(t, x) &= E[\partial_i g(X(T)) X'''_{iknm}(T) + \partial_{ir} g(X(T)) X'_{ik}(T) X''_{rnm}(T) \\ &\quad + \partial_{ir} g(X(T)) X'_{in}(T) X''_{rkm}(T) + \partial_{ir} g(X(T)) X'_{im}(T) X''_{rkn}(T) \\ &\quad + \partial_{irv} g(X(T)) X'_{ik}(T) X'_{rn}(T) X'_{vm}(T) \mid X'''_{iknm}(t) = X''_{ikn}(t) = 0, X'_{ij}(t) = \delta_{ij}, X(t) = x], \end{aligned}$$

and for the fourths derivatives:

$$(2.21) \quad \partial_{knmp} u(t, x) = \dots$$

Let $Y = (X, X', X'', X''', X'''')^T$ and let I denote the $d \times d$ identity matrix. Then the system (2.14-2.17) can be written

$$(2.22) \quad \begin{aligned} dY &= A(t, Y(t^-))dt + B^\ell(t, Y(t^-))dW^\ell(t) + \int_{\mathbf{Z}} C(t, Y(t^-), z)p(dt, dz), \quad t > t_0 \\ Y(t_0) &= (x, I, 0, 0, 0)^T. \end{aligned}$$

Furthermore, rewrite the representation (2.18-2.21) as

$$(2.23) \quad \begin{aligned} f_i(Y) &\equiv \partial_k g(X) X'_{ki}, \\ f_{ij}(Y) &\equiv \partial_k g(X) X''_{kij} + \partial_{kn} g(X) X'_{ki} X'_{nj}, \\ f_{ijm}(Y) &\equiv \partial_k g(X) X'''_{kijm} + \partial_{kn} g(X) X'_{ki} X''_{njm} \\ &\quad + \partial_{kn} g(X) X'_{kj} X''_{nim} + \partial_{kn} g(X) X'_{km} X''_{nji} \\ &\quad + \partial_{knv} g(X) X'_{ki} X'_{nj} X'_{vm}, \\ f_{ijmn}(Y) &\equiv \dots \end{aligned}$$

The Euler approximation of Y , the solution (2.22), is denoted by $\bar{Y} = (\bar{Y}^0, \bar{Y}^1, \bar{Y}^2, \bar{Y}^3, \bar{Y}^4)^T$ and can be extended as the solution of the stochastic differential equation with piecewise constant drift and diffusion fluxes

$$(2.24) \quad d\bar{Y} = \bar{A}(t; \bar{Y})dt + \bar{B}^\ell(t; \bar{Y})dW^\ell(t) + \int_{\mathbf{Z}} C(t, \bar{Y}, z)p(dt, dz),$$

defined as in (1.4), (1.6) and (2.1), with coefficients defined as in (2.2).

An important consequence of the Euler method is that the variation and the Euler discretization commute. This yields for each α the representation

$$(2.25) \quad \partial_\alpha u(t, x) - \partial_\alpha \bar{u}(t, x) = E[f_\alpha(Y(T)) - f_\alpha(\bar{Y}(T)) \mid Y(t) = \bar{Y}(t) = (x, I, 0, 0, 0)^T].$$

Now we rely on Lemma 2.1 applied to the process Y to obtain the representation

$$(2.26) \quad \begin{aligned} &E[f_\alpha(Y(T)) - f_\alpha(\bar{Y}(T)) \mid Y(t) = \bar{Y}(t) = (x, I, 0, 0, 0)^T] \\ &= \int_t^T E\left[(A - \bar{A})_k \partial_k \nu^\alpha(s, \bar{Y}(s^-)) + (D - \bar{D})_{kn} \partial_{kn} \nu^\alpha(s, \bar{Y}(s^-)) \mid \bar{Y}(t) = (x, I, 0, 0, 0)^T\right] ds \end{aligned}$$

to be used in Lemmas 2.6 and 2.8 below, where the cost to go function ν^α is, for each f_α in (2.23), defined as

$$\nu^\alpha(t, y) = E[f_\alpha(Y(T)) \mid Y(t) = y],$$

we use the notation

$$D_{kn} = \frac{1}{2} B_k^\ell B_n^\ell, \quad \bar{D}_{kn} = \frac{1}{2} \bar{B}_k^\ell \bar{B}_n^\ell,$$

and the corresponding Kolmogorov Backward equation is

$$\begin{aligned} \mathcal{L}^Y \nu^\alpha &\equiv \partial_t \nu^\alpha + A_k \partial_k \nu^\alpha + D_{kn} \partial_{kn} \nu^\alpha + \lambda(t) \int_{\mathbf{Z}} [\nu^\alpha(t, Y + C(t, Y, z)) - \nu^\alpha(t, Y)] \mu(t, dz) = 0, \quad t < T, \\ \nu^\alpha(T, \cdot) &= f_\alpha. \end{aligned}$$

We are ready for the second step.

Lemma 2.6 (Approximation of $\partial_\alpha u$). *Let the piecewise constant mesh function Δt be defined by*

$$\Delta t(s) \equiv \Delta t_n \quad \text{for } s \in [t_n, t_{n+1}) \quad \text{and } n = 0, \dots, N_A(\omega) - 1.$$

Suppose that the assumptions in Lemma 2.1 hold. Then the discretization errors of the stochastic flows, for $|\alpha| \leq 4$, satisfy

$$(2.27) \quad \partial_\alpha(u - \bar{u})(t_n, \bar{X}(t_n)) = \int_{t_n}^T E[\mathcal{O}(\Delta t(s)) \mid \mathcal{F}_{t_n}] ds = \mathcal{O}(\Delta \tilde{t}_{max}).$$

Furthermore

$$(2.28) \quad \begin{aligned} E[(a_i(t_{n+1}, \bar{X}(t_{n+1}^-)) - \bar{a}_i(t_n; \bar{X}))(\partial_i u(t_{n+1}, \bar{X}(t_{n+1}^-)) - \partial_i \bar{u}(t_{n+1}, \bar{X}(t_{n+1}^-))) \mid \mathcal{G}] \\ = \Delta t_n \int_{t_{n+1}}^T \mathcal{O}(\Delta t(s)) ds, \end{aligned}$$

$$(2.29) \quad \begin{aligned} E[(d_{ij}(t_{n+1}, \bar{X}(t_{n+1}^-)) - \bar{d}_{ij}(t_n; \bar{X}))(\partial_{ij} u(t_{n+1}, \bar{X}(t_{n+1}^-)) - \partial_{ij} \bar{u}(t_{n+1}, \bar{X}(t_{n+1}^-))) \mid \mathcal{G}] \\ = \Delta t_n \int_{t_{n+1}}^T \mathcal{O}(\Delta t(s)) ds. \end{aligned}$$

Proof. The combination of (2.25) and (2.26) give

$$(2.30) \quad \partial_\alpha(u - \bar{u})(t_n, \bar{X}(t_n)) = \int_{t_n}^T E[(A_i - \bar{A}_i)\partial_i \nu^\alpha(s, \bar{Y}(s^-)) + (D_{ij} - \bar{D}_{ij})\partial_{ij} \nu^\alpha(s, \bar{Y}(s^-)) \mid \mathcal{F}_{t_n}] ds$$

Now, for $t_m \leq t < t_{m+1}$, introduce the notation

$$h(t, \bar{Y}(t)) \equiv (A_i - \bar{A}_i)\partial_i \nu^\alpha(t, \bar{Y}(t)) + (D_{ij} - \bar{D}_{ij})\partial_{ij} \nu^\alpha(t, \bar{Y}(t)),$$

and

$$\begin{aligned} \mathcal{L}_0 w(t, y) &\equiv (\partial_t w + \bar{A}_n \partial_n w + \bar{D}_{kn} \partial_{kn} w)(t, y) \\ \mathcal{L}^{\bar{Y}} w(t, y) &\equiv \mathcal{L}_0 w(t, y) + \lambda(t) \int_{\mathbf{Z}} [w(t, y + C(t, y, z)) - w(t, y)] \mu(t, dz). \end{aligned}$$

Observe that $h(t_m, \bar{Y}(t_m)) = 0$, apply Itô's formula in the interval $[t_m, t]$ to obtain

$$E[h(t, \bar{Y}(t^-)) \mid \mathcal{F}_{t_m}] = \int_{t_m}^t E[\mathcal{L}^{\bar{Y}} h(s, \bar{Y}(s^-)) \mid \mathcal{F}_{t_m}] ds = \mathcal{O}(\Delta t_m), \quad t_m \leq t < t_{m+1},$$

where the bound follows as in (2.13). This plugged into (2.30) proves (2.27).

The estimate (2.29) follows similarly by defining now

$$\tilde{h}(t, \bar{X}(t)) \equiv (d_{ij} - \bar{d}_{ij})\partial_{ij}(u - \bar{u})(t, \bar{X}(t)).$$

Then the Itô's formula shows as in (2.12-2.13)

$$E[\tilde{h}(t, \bar{X}(t)) \mid \mathcal{G}] = \int_{t_n}^t E[\mathcal{L}_0 \tilde{h}(s, \bar{X}(s^-)) \mid \mathcal{G}] ds, \quad t_n \leq t < t_{n+1}.$$

The final step to prove (2.29) is to establish

$$E[\mathcal{L}_0 \tilde{h}(s, \bar{X}(s^-)) \mid \mathcal{G}] = \int_s^T \mathcal{O}(\Delta t(\tau)) d\tau.$$

The function $\mathcal{L}_0 \tilde{h}(s, \bar{X}(s^-))$ splits into the two types of terms $h_1 \equiv (d - \bar{d})v$ and $h_2 \equiv v\partial_\alpha(u - \bar{u})$, with smooth functions v of $(s, \bar{X}(s))$. The Itô formula again shows that

$$E[h_1(s, \bar{X}(s^-)) \mid \mathcal{G}] = \int_{t_n}^s E[\mathcal{L}_0 h_1(\tau, \bar{X}(\tau^-)) \mid \mathcal{G}] d\tau = \mathcal{O}(\Delta t_n), \quad t_n \leq s < t_{n+1}.$$

Moreover (2.27) implies

$$E[h_2(s, \bar{X}(s^-)) \mid \mathcal{G}] = \int_s^T \mathcal{O}(\Delta t(\tau)) d\tau,$$

and consequently (2.29) holds. The estimate (2.28) of the drift terms follows analogously. \square

Definition 2.7 (Local discrete solution operator). *Write the Euler time stepping for the time nodes $t = t_0, t_1^-, t_1, t_2^-, t_2, \dots$ as*

$$(2.31) \quad \bar{X}(t_{next}, \omega) = \Phi(t, \bar{X}(t), \omega)$$

where the next computation time is given by

$$t_{next} = \begin{cases} t_n & \text{if } t = t_n^- \\ t_{n+1}^- & \text{if } t = t_n. \end{cases}$$

Observe that depending on ω and t , we may have

$$\Phi(t, x, \omega) = \begin{cases} x + a(t, x)\Delta t + b^\ell(t, x)\Delta W^\ell, & \text{if } t = t_n, \\ x + c(t, x, Z_k(\omega)), & \text{if } t = t_n^- \text{ and } t_n \text{ is the } k\text{-th jump time,} \\ x & \text{if } t = t_n^- \text{ and } t_n \text{ is not a jump time.} \end{cases}$$

Lemma 2.8 (Representation with discrete duals). *Suppose that the assumptions in Lemma 2.1 hold. Then the dual functions φ and φ' , defined Appendix A, satisfy for $t = t_n$ and $t = t_{n+1}^-$*

$$(2.32) \quad \partial_i \bar{u}(t, \bar{X}(t)) = E[\varphi_i(t) \mid \mathcal{F}_t],$$

$$(2.33) \quad \partial_{ij} \bar{u}(t, \bar{X}(t)) = E[\varphi'_{ij}(t) \mid \mathcal{F}_t].$$

Proof. Equations (2.1), (2.14) and (2.18) show that the first variation of the Euler approximation \bar{X} is in fact equal to the Euler approximation of the first variation X' and consequently

$$\partial_i \bar{u}(t, \bar{X}(t)) = E[\partial_j g(\bar{X}(T)) \bar{X}'_{ji}(T; t) \mid \mathcal{F}_t],$$

where $\bar{X}'_{ji}(s; t)$ ($s > t$), is the Euler approximation (2.24) of X' with initial data $\bar{X}'_{ji}(t; t) = \delta_{ji}$.

Let, for $t = \dots, t_n, t_{n+1}^-, \dots$,

$$\varphi_i(t) \equiv \partial_{x_i(t)} g(\bar{X}(T)),$$

i.e.

$$(2.34) \quad \varphi_i(t) = \partial_j g(\bar{X}(T)) \bar{X}'_{ji}(T; t).$$

We prove inductively that $\varphi_i(t)$ is the solution of the corresponding problem in (A.1)-(A.4). Since (A.1) is trivially true it remains to prove the inductive step. By the chain rule we have

$$\partial_k g(\bar{X}(T)) \bar{X}'_{ki}(T; t) = \partial_k g(\bar{X}(T)) \bar{X}'_{kj}(T; t_{next}) \bar{X}'_{ji}(t_{next}; t)$$

or in other words

$$(2.35) \quad \varphi_i(t) = \varphi_j(t_{next}) \partial_i \Phi_j(t, \bar{X}(t))$$

which is equivalent to (A.1)-(A.4), what we wanted to prove.

The equality (2.34) implies that

$$\partial_{ij} \bar{u}(t, \bar{X}(t)) = E[\partial_{x_j(t)} \varphi_i(t) \mid \mathcal{F}_t].$$

The next step is to verify that the first variation of φ ,

$$(2.36) \quad \varphi'_{ij}(t) \equiv \partial_{x_j(t)} \varphi_i(t) \equiv \frac{\partial \varphi_i(t; \bar{X}(t) = x)}{\partial x_j},$$

satisfies the backward recursive equation (A.1)-(A.4). First, differentiate the equation (2.35) to obtain

$$(2.37) \quad \begin{aligned} \varphi'_{ik}(t) &= \partial_i \Phi_j \partial_{x_k(t)} \varphi_j(t_{next}) + \partial_k \partial_i \Phi_j \varphi_j(t_{next}), \quad t < T, \\ \varphi'_{ik}(T) &= \partial_{ik} g(\bar{X}(T)). \end{aligned}$$

Observe that the problem (A.1)-(A.4) shows that $\varphi(t_{next})$ depends only on the point values

$$\{\bar{X}(s) : t_{next} \leq s \leq T\},$$

so that

$$(2.38) \quad \partial_{x_k(t)} \varphi_j(t_{next}) = \partial_{x_p(t_{next})} \varphi_j(t_{next}) \partial_{x_k(t)} \bar{X}_p(t_{next}).$$

Finally, the definitions of \bar{X} and Φ in (2.1) and (2.31) imply

$$(2.39) \quad \partial_k \Phi_p(t, \bar{X}(t)) = \partial_{x_k(t)} \bar{X}_p(t_{next}),$$

which together with (2.37-2.38) prove that φ' satisfies the recursive equation

$$(2.40) \quad \begin{aligned} \varphi'_{ik}(t) &= \partial_i \Phi_j \quad \varphi'_{jp}(t_{next}) \partial_k \Phi_p(t, \overline{X}(t)) + \partial_k \partial_i \Phi_j \quad \varphi_j(t_{next}), \quad t < T, \\ \varphi'_{ik}(T) &= \partial_{ik} g(\overline{X}(T)), \end{aligned}$$

which is equivalent to (A.1)-(A.4). \square

Remark 2.9. The measurability of $(a_i(t_{n+1}^-, \overline{X}(t_{n+1}^-)) - \overline{a}_i(t_n, \overline{X}(t_n))) \in \mathcal{F}_{t_{n+1}^-}$ proves that for $t = t_{n+1}^-$ and any random variable β we have

$$\begin{aligned} E[(a_i(t, \overline{X}(t)) - \overline{a}_i(t_n, \overline{X}(t_n))) E[\beta | \mathcal{F}_t]] &= E[E[(a_i(t, \overline{X}(t)) - \overline{a}_i(t_n, \overline{X}(t_n))) \beta | \mathcal{F}_t]] \\ &= E[(a_i(t, \overline{X}(t)) - \overline{a}_i(t_n, \overline{X}(t_n))) \beta], \end{aligned}$$

and in a completely similar way we obtain the same result for the diffusion terms, i.e.

$$E[(d_{ij}(t, \overline{X}(t)) - \overline{d}_{ij}(t_n, \overline{X}(t_n))) E[\beta | \mathcal{F}_t]] = E[(d_{ij}(t, \overline{X}(t)) - \overline{d}_{ij}(t_n, \overline{X}(t_n))) \beta].$$

The proof of Theorem 2.3 is now concluded by combining Lemmas 2.5, 2.6, 2.8, Remark 2.9, and the Central Limit Theorem to estimate I_M and II_M . We have

$$\begin{aligned} & \int_{\tilde{t}_m}^{\tilde{t}_{m+1}} E[(a_i(t, \overline{X}(t^-)) - \overline{a}_i(t, \overline{X})) \partial_i u(t, \overline{X}(t^-))] dt \\ &= E\left[\sum_{n \in \mathcal{J}_m} (a_i(t_{n+1}, \overline{X}(t_{n+1}^-)) - a_i(t_n, \overline{X}(t_n))) \partial_i u(t_{n+1}, \overline{X}(t_{n+1}^-)) \frac{\Delta t_n}{2} \right] \\ & \quad + \sum_{m=0}^{N-1} \mathcal{O}((\Delta \tilde{t}_m)^3) \\ &= E\left[\sum_{n \in \mathcal{J}_m} (a_i(t_{n+1}, \overline{X}(t_{n+1}^-)) - a_i(t_n, \overline{X}(t_n))) \partial_i \bar{u}(t_{n+1}, \overline{X}(t_{n+1}^-)) \frac{\Delta t_n}{2} \right] \\ & \quad + \sum_{m=0}^{N-1} E\left[\sum_{n \in \mathcal{J}_m} (\Delta t_n)^2 \int_{t_{n+1}}^T \mathcal{O}(\Delta t(s)) ds \right] + \sum_{m=0}^{N-1} \mathcal{O}((\Delta \tilde{t}_m)^3) \\ &= E\left[\sum_{n \in \mathcal{J}_m} (a_i(t_{n+1}, \overline{X}(t_{n+1}^-)) - a_i(t_n, \overline{X}(t_n))) E[\varphi_i(t_{n+1}-) | \mathcal{F}_{t_{n+1}^-}] \frac{\Delta t_n}{2} \right] \\ & \quad + \sum_{m=0}^{N-1} E\left[\sum_{n \in \mathcal{J}_m} (\Delta t_n)^2 \int_{t_{n+1}}^T \mathcal{O}(\Delta t(s)) ds \right] + \sum_{m=0}^{N-1} \mathcal{O}((\Delta \tilde{t}_m)^3) \\ &= E\left[\sum_{n \in \mathcal{J}_m} (a_i(t_{n+1}, \overline{X}(t_{n+1}^-)) - a_i(t_n, \overline{X}(t_n))) \varphi_i(t_{n+1}-) \frac{\Delta t_n}{2} \right] \\ & \quad + \sum_{m=0}^{N-1} E\left[\sum_{n \in \mathcal{J}_m} (\Delta t_n)^2 \int_{t_{n+1}}^T \mathcal{O}(\Delta t(s)) ds \right] + \sum_{m=0}^{N-1} \mathcal{O}((\Delta \tilde{t}_m)^3) \end{aligned}$$

The expansion of the diffusion term appearing in (2.9) follows analogously. \square

Observe that the number of realizations to determine a reliable error estimate is in general TOL^{-1} , much smaller than the, proportional to TOL^{-2} , number of realizations to approximate $E[g(X(T))]$. For more details on this and the statistical approximation of the error density ρ see Remark 2.7 in [34].

3. AN ERROR ESTIMATE WITH STOCHASTIC TIME STEPS

This section derives error estimates with time steps which are stochastic and determined individually for each realization by the whole solution path \overline{X} . The analysis will use the Malliavin derivative, $\partial_{W(t)} Y$, which is the first variation of a process Y with respect to a perturbation $dW(t)$, at time t of the Wiener

process, cf. [30]. The Malliavin derivative for a stochastic integral X is related to the first variation, $\partial_{x(t)}\bar{X}$, for a perturbation of the position at time t by

$$(3.1) \quad \begin{aligned} \partial_{W^\ell(t)}X(\tau) &= \frac{\partial X_k(t)}{\partial W^\ell(t)} \partial_{x_k(t)}X(\tau) = b_k^\ell(X(t)) \partial_{x_k(t)}X(\tau), \quad \tau > t, \\ \partial_{W^\ell(t)}X(\tau) &= 0 \quad \tau < t, \end{aligned}$$

if $dX_k = a_k(X(t))dt + b_k^\ell(X(t))dW^\ell$ (cf. (2.7)).

We shall restrict the analysis to time steps which are constructed by first sampling the jump times to augment an a priori given time-discretization Δt , obtaining $\Delta t[0]$ (see (3.6)) and then using the refinement criterion

$$(3.2) \quad \begin{aligned} \Delta t(t) &= \Delta t[0](t)/2^n, \quad \text{for some natural number } n = n(t, \omega), \\ |\rho(t, \omega)|(\Delta t(t))^2 &< \text{constant}, \end{aligned}$$

with an approximate error density function, ρ , satisfying, for $s \in [0, T]$, $t \in [0, T]$ and all outcomes ω , the uniform upper and lower bounds

$$(3.3) \quad \begin{aligned} c(\text{TOL}) &\leq |\rho(s, \omega)| \leq C(\text{TOL}), \\ |\partial_{W(t)}\rho(s, \omega)| &\leq C(\text{TOL}), \end{aligned}$$

for some positive functions c and C , with $\text{TOL}/c(\text{TOL}) \rightarrow 0$ as $\text{TOL} \rightarrow 0$. For each realization successive subdivisions of the steps yield the largest time steps satisfying (3.2). The corresponding stochastic increments ΔW will have the correct distribution, with the necessary independence, if the increments ΔW related to the new steps are generated by Brownian bridges, cf. [22], i.e. the time steps are generated by conditional expected values of the Wiener process.

Let δ be a constant approximating $\frac{\text{TOL}}{E[N]}$, where $E[N]$ is the expected number of steps. The analysis in this section with adaptive non adapted time steps, satisfying (3.2)-(3.3), is based on the following Stochastic time step algorithm described in next page.

Section 4.4 presents a more precise formulation of this algorithm. Lemma 3.1 and Theorem 3.3 below show that although the steps generated by (3.2)-(3.3) through the algorithm above are not adapted, the method indeed converges to the correct limit as the forward Euler method with adapted time steps.

Lemma 3.1 (Strong convergence). *Suppose that a, b, g, X satisfy the assumptions in Lemma 2.1 and that \bar{X} is constructed by the forward Euler method, based on the stochastic time step algorithm above, with step sizes Δt_n satisfying (3.2-3.3) and their corresponding ΔW_n are generated by Brownian bridges. Assume also that $\bar{X}(0) = X(0)$. Then*

$$\sup_{0 \leq t \leq T} \sqrt{E[|X(t) - \bar{X}(t)|^2]} = \mathcal{O}(\sqrt{\Delta t_{sup}}) = \mathcal{O}\left(\sqrt{\frac{\text{TOL}}{c(\text{TOL})}}\right) \rightarrow 0,$$

as $\text{TOL} \rightarrow 0$, where $\Delta t_{sup} \equiv \sup_{n, \omega} \Delta t_n(\omega)$.

Proof. Let \mathcal{G} be the σ -algebra generated by the jumps and marks constructed in Remark 1.1. Consider the conditional expectation $E[|X(t) - \bar{X}(t)|^2 | \mathcal{G}]$ and apply Lemma 3.1 from [34], using also that there is no time discretization error at the jump nodes. \square

In addition to the dual functions φ and φ' in Theorem 2.3, the new error expansion for stochastic time steps in Theorem 3.3 below also uses, for $t = t_{n+1}^-$, the discrete dual variation

$$(3.4) \quad \varphi''_{ikm}(t) \equiv \partial_{x_m(t)}\varphi'_{ik}(t) \equiv \frac{\partial \varphi'_{ik}(t; \bar{X}(t)=x)}{\partial x_m},$$

which satisfies the backward problem (A.1) – (A.4), i.e.,

Lemma 3.2. Let Φ , φ and φ' be defined by (2.31) and (A.1) – (A.4). Then φ'' is given by

$$\begin{aligned}
(3.5) \quad \varphi''_{ikm}(t) = & \partial_i \Phi_j(t, \overline{X}(t)) \partial_k \Phi_p(t, \overline{X}(t)) \partial_m \Phi_r(t, \overline{X}(t)) \varphi''_{jpr}(t_{next}) \\
& + \partial_{im} \Phi_j(t, \overline{X}(t)) \partial_k \Phi_p(t, \overline{X}(t)) \varphi'_{jp}(t_{next}) \\
& + \partial_i \Phi_j(t, \overline{X}(t)) \partial_{km} \Phi_p(t, \overline{X}(t)) \varphi'_{jp}(t_{next}) \\
& + \partial_{ik} \Phi_j(t, \overline{X}(t)) \partial_m \Phi_p(t, \overline{X}(t)) \varphi'_{jp}(t_{next}) \\
& + \partial_{ikm} \Phi_j(t, \overline{X}(t)) \varphi_j(t_{next}), \quad t < T, \\
\varphi''_{ikm}(T) = & \partial_{ikm} g(\overline{X}(T)).
\end{aligned}$$

which is equivalent to (A.1) – (A.4) with t_{next} as in Definition 2.7.

Stochastic time step algorithm:

Do for M realizations ω_j , $j = 1, \dots, M$:

Sample jump times: $0 < \tau_1(\omega_j) < \dots < \tau_{\widehat{N}(\omega_j)}(\omega_j) < T$.

Consider an augmented partition given by the union

$$(3.6) \quad \Delta t[0] = \{t_n(\omega_j)\}_{n=0}^{N_A(\omega_j)} = \{\tilde{t}_n\}_{n=0}^N \cup \{\tau_n(\omega_j)\}_{n=0}^{\widehat{N}(\omega_j)}$$

that has $N_A(\omega_j) = N + \widehat{N}(\omega_j)$ (a.s.) time steps.

STEP 1: Set $k = 0$. Start with the initial coarse mesh $\Delta t[0]$ and compute $\Delta W[0]$.

STEP 2: For the piecewise constant mesh function $\Delta t[k]$ with corresponding noise $\Delta W[k]$, compute $\overline{X}[k]$ and the weight function $\rho[k]$ defined in (3.8-3.9).

STEP 3: Define $r(t) \equiv |\rho[k](t)|(\Delta t[k](t))^2$ and let for all t

$$\Delta t[k+1](t) = \begin{cases} \Delta t[k](t), & \text{if } r(t) < \delta, \quad (\dagger) \\ \Delta t[k](t)/2, & \text{if } r(t) \geq \delta, \quad (\star) \end{cases}$$

and in the refinement case (\star) construct $\Delta W[k+1]$ by Brownian bridges based on the already known $\Delta W[k]$.

STEP 4: **If** at least one step of $\Delta t[k]$ is refined by (\star) , increment k by 1 and **goto** Step 2. **Else** all steps of $\Delta t[k]$ satisfy (\dagger) and accept the approximation $g(\overline{X}(T, \omega_j))$ and **goto** the next realization of p and W .

Enddo

If the statistical error, $E[g(\overline{X}(T))] - \frac{1}{M} \sum_{j=1}^M g(\overline{X}(T, \omega_j))$, is sufficiently small

stop, **else** restart with a larger M . **Endif**

Proof. Differentiation of the backward recursive equation (2.40) and the relations (2.34-2.39) together with

$$(3.7) \quad \partial_{x_m(t)} \varphi'_{jp}(t_{next}) = \partial_{x_r(t_{next})} \varphi'_{jp}(t_{next}) \partial_{x_m(t)} \overline{X}_r(t_{next}),$$

prove as in (2.36)-(2.39) that φ'' satisfies (3.5). Here, (3.7) holds since the linear system for the variable $(\varphi(t_{next}), \varphi'(t_{next}))$ depends only on the point values $\{\overline{X}(s) : t_{next} \leq s \leq T\}$. \square

The following theorem derives an error estimate applicable both to adaptive deterministic time steps and to the stochastic time step algorithm; the assumptions and the proof of the theorem focus on stochastic steps, however a modification to deterministic time steps is straightforward. The computable error density $|\tilde{\rho}|$ of this error estimate can then be cut-off for small and large values to satisfy (3.3), see (4.17) and (4.6).

Theorem 3.3 (Stochastic time steps error expansion). *Suppose that a, b, g, X satisfy the assumptions in Lemma 2.1 and that \overline{X} is constructed by the forward Euler method with step sizes Δt_n satisfying (3.2-3.3) and the corresponding ΔW_n are generated by Brownian bridges, following the stochastic time step algorithm in Lemma 3.1. Assume also that $\overline{X}(0) = X(0)$ and $E[|X(0)|^{k_0}] \leq C$ for some $k_0 \geq 16$. Then*

the time discretization error has the following expansion, based on both the drift and diffusion fluxes and the discrete dual functions φ , φ' and φ'' given in (A.1) – (A.4), with computable leading order terms

$$(3.8) \quad E[g(X(T)) - g(\overline{X}(T))] = E\left[\sum_{n=0}^{N_A-1} \tilde{\rho}(t_n, \overline{X})(\Delta t_n)^2\right] + \mathcal{O}\left(\sqrt{\frac{\text{TOL}}{c(\text{TOL})}} \left(\frac{C(\text{TOL})}{c(\text{TOL})}\right)^{8/k_0}\right) E\left[\sum_{n=0}^{N_A-1} (\Delta t_n)^2\right],$$

where

$$(3.9) \quad \begin{aligned} \tilde{\rho}(t_n, \overline{X}) \equiv & \frac{1}{2} \left((\partial_t a_k + \partial_j a_k a_j + \partial_{ij} a_k d_{ij}) \varphi_k(t_{n+1}^-) \right. \\ & + (\partial_t d_{km} + \partial_j d_{km} a_j + \partial_{ij} d_{km} d_{ij} + 2\partial_j a_k d_{jm}) \varphi'_{km}(t_{n+1}^-) \\ & \left. + (2\partial_j d_{km} d_{jr}) \varphi''_{kmr}(t_{n+1}^-) \right), \end{aligned}$$

and the terms in the sum of (3.9) are evaluated at the a posteriori known points $(t_n, \overline{X}(t_n))$, i.e.

$$\begin{aligned} \partial_\alpha a &\equiv \partial_\alpha a(t_n, \overline{X}(t_n)), \\ \partial_\alpha b &\equiv \partial_\alpha b(t_n, \overline{X}(t_n)), \\ \partial_\alpha d &\equiv \partial_\alpha d(t_n, \overline{X}(t_n)). \end{aligned}$$

Proof. We consider the difference

$$g(X(t)) - g(\overline{X}(t))$$

and apply Theorem 3.3 from [34], using also that there is no time discretization error at the jump nodes. To this end, denote the set of stochastic time nodes by $\mathcal{J} \equiv \{0 = t_0, t_1^-, t_1, t_2^-, t_2, \dots, t_N = T\}$ and recall that the notation

$$(3.10) \quad \overline{X}(t_{next}) = \Phi(\overline{X}(t)), \quad t \in \mathcal{J},$$

introduced in Definition 2.7 denotes one step with the Euler method. Write similarly one step with the exact solution

$$(3.11) \quad X(t_{next}) = \hat{\Phi}(X(t)), \quad t \in \mathcal{J}.$$

Introduce the notation $X^t \equiv X(t)$ and $\overline{X}^t \equiv \overline{X}(t)$. Now verify the representation

$$(3.12) \quad g(X(T)) - g(\overline{X}(T)) = \sum_{n=0}^{N_A-1} (\hat{\Phi}(\overline{X}(t_n)) - \Phi(\overline{X}(t_n)))_i \tilde{\varphi}_i(t_{n+1}^-)$$

where the weight functions are defined recursively by the linear backward recursion

$$(3.13) \quad \begin{aligned} \tilde{\varphi}_i(T) &= \int_0^1 \partial_i g(sX(T) + (1-s)\overline{X}(T)) ds, \\ \tilde{\varphi}_i(t) &= \left(\int_0^1 \partial_i \hat{\Phi}_j(sX(t) + (1-s)\overline{X}(t)) ds \right) \tilde{\varphi}_j(t_{next}), \quad t \in \mathcal{J}. \end{aligned}$$

To verify (3.12), first observe that by construction of the Euler method at every jump point $t = t_n^-$ there is no local error in computing the next \overline{X} value at time t_n , i.e.

$$\hat{\Phi}(\overline{X}^t) - \Phi(\overline{X}^t) = 0$$

so (3.12) is equivalent to

$$(3.14) \quad g(X(T)) - g(\overline{X}(T)) = \sum_{t \in \mathcal{J}} (\hat{\Phi}(\overline{X}^t) - \Phi(\overline{X}^t))_i \tilde{\varphi}_i(t_{next}).$$

Then telescoping cancelation gives

$$(3.15) \quad g(X(T)) - g(\overline{X}(T)) = \sum_{t \in \mathcal{J}} ((X^{t_{next}} - \overline{X}^{t_{next}})_i \tilde{\varphi}_i(t_{next}) - (X^t - \overline{X}^t)_i \tilde{\varphi}_i(t)).$$

Use the definitions (3.10,3.11) and split the first term in the sum of (3.15) into

$$(\hat{\Phi}(X^t) - \hat{\Phi}(\bar{X}^t) + \hat{\Phi}(\bar{X}^t) - \Phi(\bar{X}^t))_i \tilde{\varphi}_i(t_{next}).$$

The two first terms above and the last term in the sum of (3.15) combine to zero by (3.13):

$$(\hat{\Phi}(X^t) - \hat{\Phi}(\bar{X}^t))_i \tilde{\varphi}_i(t_{next})(X^t - \bar{X}^t)_i \tilde{\varphi}_i(t),$$

which proves (3.12).

The next step is to use the Malliavin derivative to analyze the expectation of the representation (3.12) by studying the dependence of \bar{X} and $\tilde{\varphi}$ on a small increment dW . This follows exactly the lines of the proof of Theorem 3.3 from [34], and it is not reproduced here. \square

4. ADAPTIVE TIME-STEPPING ALGORITHMS

Here, we describe two adaptive time-stepping algorithms for the weak approximation problem of (1.1) based on the approximation error formulas described in the previous section. They are very similar to those introduced in [29]. Algorithm-D is based on a quasi-deterministic mesh that is fixed for all realizations and its adaptive strategy is based on averaged information from the a posteriori error formula. On the other hand, Algorithm-S can adapt the time discretization differently for each realization. Proper sample of the Wiener process is possible by means of Brownian bridges. Both adaptive algorithms choose adaptively the number of realizations and the size of time steps to efficiently bound the approximation error by a prescribed error tolerance.

4.1. Computational error splitting. The weak approximation computational error of the Monte Carlo Euler method, $\mathcal{E}_C \equiv E[g(X(T))] - \frac{1}{M} \sum_{j=1}^M g(\bar{X}(T, \omega_j))$, naturally separates to the time discretization error $\mathcal{E}_T \equiv E[g(X(T))] - E[g(\bar{X}(T))]$ and the statistical error $\mathcal{E}_S \equiv E[g(\bar{X}(T))] - \frac{1}{M} \sum_{j=1}^M g(\bar{X}(T, \omega_j))$, i.e., $\mathcal{E}_C = \mathcal{E}_T + \mathcal{E}_S$. Thus, the control the computational error is related to the a combined control of the time discretization error via the choice of the time steps Δt and of the statistical error via the choice of the number M of the realizations. Therefore, we split a given computational error tolerance, $\text{TOL} > 0$, into a statistical error tolerance TOL_S and a time discretization error tolerance TOL_T (see [34], [29]) by

$$(4.1) \quad \text{TOL}_T = \frac{1}{3} \text{TOL} \quad \text{and} \quad \text{TOL}_S = \frac{2}{3} \text{TOL}.$$

4.2. Control of the statistical error. For M independent samples $\{Y(\omega_j)\}_{j=1}^M$ of a random variable Y , with $E[|Y|^6] < \infty$, define the sample average by

$$\mathcal{A}(Y; M) \equiv \frac{1}{M} \sum_{j=1}^M Y(\omega_j)$$

and the sample standard deviation by

$$\mathcal{S}(Y; M) \equiv \{\mathcal{A}(Y^2; M) - (\mathcal{A}(Y; M))^2\}^{\frac{1}{2}}.$$

Let $\sigma_Y \equiv \{E[|Y - E[Y]|^2]\}^{\frac{1}{2}}$ and $Z_M \equiv \frac{\sqrt{M}}{\sigma_Y} (\mathcal{A}(Y; M) - E[Y])$ with cumulative distribution function $F_{Z_M}(x) \equiv P(Z_M \leq x)$, $x \in \mathbb{R}$. Let $\lambda \equiv (E[|Y - E[Y]|^3])^{1/3} / \sigma_Y < \infty$, then the Berry-Esseen theorem, cf. [10], gives the following estimation in the Central Limit Theorem

$$\sup_{x \in \mathbb{R}} |F_{Z_M}(x) - \mathcal{N}(x)| \leq \frac{3}{\sqrt{M}} \lambda^3$$

for the rate of convergence of F_{Z_M} to the distribution function \mathcal{N} of a normal random variable with mean zero and variance one, i.e.

$$(4.2) \quad \mathcal{N}(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}s^2\right) ds.$$

Since in the examples below M is sufficiently large, i.e. $M \gg 36\lambda^6$, the statistical error

$$\mathcal{E}_S(Y; M) \equiv E[Y] - \mathcal{A}(Y; M)$$

satisfies, by the Berry-Esseen theorem, the following probability approximation

$$P\left(\left[|\mathcal{E}_S(Y; M)| \leq c_0 \frac{\sigma_Y}{\sqrt{M}}\right]\right) \simeq 2\mathcal{N}(c_0) - 1.$$

In practice choose some constant $c_0 \geq 1.65$, so the normal distribution satisfies $1 > 2\mathcal{N}(c_0) - 1 \geq 0.901$ and the event

$$(4.3) \quad |\mathcal{E}_S(Y; M)| \leq \mathbf{E}_S(Y; M) \equiv c_0 \frac{\mathcal{S}(Y; M)}{\sqrt{M}}$$

has probability close to one, which involves the additional step to approximate σ_Y by $\mathcal{S}(Y; M)$, cf. [13]. Thus, in the computations $\mathbf{E}_S(Y; M)$ is a good approximation of the statistical error $\mathcal{E}_S(Y; M)$.

For a given $\text{TOL}_S > 0$, the goal is to find M such that $\mathbf{E}_S(Y; M) \leq \text{TOL}_S$. The following algorithm adaptively finds the number of realizations M to compute the sample average $\mathcal{A}(Y; M)$ as an approximation to $E[Y]$. With probability close to one, depending on c_0 , the statistical error in the approximation is then bounded by TOL_S . For technical reasons (see [29]) we choose $M = 2^n$, $n \in \mathbb{N}$.

$$(4.4) \quad \begin{aligned} &\text{routine Monte-Carlo}(\text{TOL}_S, M_0; EY) \\ &\quad \text{Set the batch counter } m = 1, M[m] = M_0 \text{ and } \mathbf{E}_S[m] = +\infty. \\ &\quad \text{Do while } (\mathbf{E}_S[m] > \text{TOL}_S) \\ &\quad \quad \text{Compute } M[m] \text{ new samples of } Y, \text{ along with the sample} \\ &\quad \quad \text{average } EY \equiv \mathcal{A}(Y; M[m]), \text{ the sample standard deviation} \\ &\quad \quad \mathcal{S}[m] \equiv \mathcal{S}(Y; M[m]) \text{ and the statistical error estimation} \\ &\quad \quad \mathbf{E}_S[m+1] \equiv \mathbf{E}_S(Y, M[m]). \text{ Compute } M[m+1] \text{ by} \\ &\quad \quad \text{change_M } (M[m], \mathcal{S}[m], \text{TOL}_S; M[m+1]). \\ &\quad \quad \text{Increase } m \text{ by 1.} \end{aligned}$$

end-do
end of Monte-Carlo

$$(4.5) \quad \begin{aligned} &\text{routine change_M } (M_{in}, \mathcal{S}_{in}, \text{TOL}_S; M_{out}) \\ &\quad M^* = \min\left\{\text{integer part}\left(\frac{c_0 \mathcal{S}_{in}}{\text{TOL}_S}\right)^2, \text{MCH} \times M_{in}\right\} \\ &\quad n = \text{integer part}(\log_2 M^*) + 1 \\ &\quad M_{out} = 2^n. \end{aligned}$$

end of change_M

Here, M_0 is a given initial value for M , and $\text{MCH} > 1$ is a positive integer parameter introduced to avoid a large new number of realizations in the next batch due to a possibly inaccurate sample standard deviation $\mathcal{S}[m]$. Indeed, $M[m+1]$ cannot be greater than $\text{MCH} \times M[m]$.

4.3. Deterministic time stepping algorithm. Following closely [29] and [27], we present an adaptive algorithm based on cut-off of the error density $\tilde{\rho}$ in (3.9) of Theorem 3.3, ρ_D , which is defined as

$$(4.6) \quad \rho_D^n \equiv \min\left(\max\left(\left|\frac{1}{(\Delta t_n)^2} \sum_{\ell \in \mathcal{J}_n} (\Delta t_\ell)^2 \tilde{\rho}(t_\ell, \bar{X})\right|, \text{TOL}^{\frac{1}{9}}\right), \text{TOL}^{-1}\right), \quad n = 1, \dots, N.$$

The error expansion in Theorem 3.3 motivates us to approximate the time discretization error by

$$(4.7) \quad |\mathcal{E}_T| \lesssim E\left[\sum_{n=1}^N r_n\right]$$

where the error indicator, r_n , is defined by

$$(4.8) \quad r_n \equiv \rho_D^n (\Delta \tilde{t}_n)^2 \quad n = 1, \dots, N.$$

The main advantage of the deterministic time stepping algorithm over the stochastic time stepping algorithm is that the number M_T of realizations necessary to determine the optimal deterministic stepping

scheme is considerable smaller than M , whereas, in the stochastic time stepping algorithm, a refinement of the partition is carried out in each one of the M trajectories, leading to a considerable larger amount of computational work.

As pointed out before, the error expansion derived in Theorem 3.3 is also valid with deterministic time steps. Therefore, we have some flexibility in the choice of error densities for the adaptive algorithm with deterministic time steps because we can also use the results from Theorem 2.3. There are some practical differences to mention. On the one hand, the variance of the averaged error density from (2.9) is $\mathcal{O}(\frac{1}{\Delta \tilde{t}_n M_T})$. This feature has been observed in [28] and [34], where a local filtering procedure was proposed to reduce the variance of the error density estimator. A positive feature of this error density is that it does not require the computation of the second variation, φ'' , which may be computationally expensive for large d . On the other hand, the averaged error density from (3.9) has a much smaller variance $\mathcal{O}(\frac{1}{M_T})$ which does not need filtering but it requires the computation of φ'' . In this work we will not discuss further this choice and only show numerical results with adaptive deterministic time steps based on the error density (3.9).

The approximation of the time discretization error in the right hand side of (4.7) can be separated into two parts

$$(4.9) \quad E \left[\sum_{n=1}^N r_n \right] \leq \mathcal{A} \left(\sum_{n=1}^N r_n; M_T \right) + \left| E \left[\sum_{n=1}^N r_n \right] - \mathcal{A} \left(\sum_{n=1}^N r_n; M_T \right) \right|,$$

where the second error term in the right hand side of (4.9) is with probability close to one asymptotically bounded by

$$(4.10) \quad \left| E \left[\sum_{n=1}^N r_n \right] - \mathcal{A} \left(\sum_{n=1}^N r_n; M_T \right) \right| \lesssim \mathbf{E}_{TS} \equiv c_0 \frac{\mathcal{S} \left(\sum_{n=1}^N r_n; M_T \right)}{\sqrt{M_T}}$$

and the first term defines $\mathbf{E}_{TT} \equiv \mathcal{A} \left(\sum_{n=1}^N r_n; M_T \right)$. Then for a given $\text{TOL}_T > 0$, the goal is to construct a partition $\Delta \tilde{t}$ of $[0, T]$, with as few time steps and realizations M_T as possible, such that $\mathbf{E}_{TT} + \mathbf{E}_{TS} \leq \text{TOL}_T$. To this end, first split the time discretization tolerance TOL_T in two positive parts TOL_{TT} and TOL_{TS} for \mathbf{E}_{TT} and \mathbf{E}_{TS} , respectively. The statistical error of the time discretization using the density (4.6) is $\mathcal{O}(\frac{\Delta t_{sup}}{\sqrt{M_T}})$. Therefore the percentage of the tolerance, TOL , devoted to the control of the statistical time discretization error can be arbitrary small as $\Delta t_{sup} \rightarrow 0$. In practice we choose

$$(4.11) \quad \text{TOL}_{TT} = \frac{2}{3} \text{TOL}_T = \frac{2}{9} \text{TOL}, \quad \text{TOL}_{TS} = \frac{1}{3} \text{TOL}_T = \frac{1}{9} \text{TOL}.$$

The control of the statistical time discretization error determines the number of realizations M_T necessary to ensure a reliable choice of the time discretization in the deterministic time stepping algorithm. Similarly as in [29], here it is optimal to equidistribute the error contributions from different time intervals. Thus, the goal of the adaptive algorithm described below is to construct a deterministic time partition $\Delta \tilde{t}$ of $[0, T]$ such that

$$(4.12) \quad \bar{r}_n \equiv \mathcal{A}(r_n; M_T) \leq d_1 \frac{\text{TOL}_{TT}}{N}, \quad n = 1, \dots, N,$$

where $d_1 = 2$, see Remark 3.9 in [29].

To achieve (4.12), start with an initial partition $\Delta \tilde{t}[1]$ and then specify iteratively a new partition $\Delta \tilde{t}[k+1]$, from $\Delta \tilde{t}[k]$, using the following refinement strategy:

(4.13) **for** $n = 1, 2, \dots, N[k]$
(4.14) **if** $\bar{r}_n[k] \geq d_1 \frac{\text{TOL}_{TT}}{N[k]}$, **then** divide $\Delta \tilde{t}_n[k]$ into 2 uniform substeps.
 else let the new step be the same as the old
(4.15) **endif**.
 endfor.

until the following stopping criteria is satisfied:

(4.16) **if** $\left(\max_{1 \leq n \leq N[k]} \bar{r}_n[k] < D_1 \frac{\text{TOL}_{TT}}{N[k]} \right)$ **then** stop.

Here D_1 is a given constant satisfying $D_1 > \frac{2}{c}d_1$ where $c \approx 1/2$, see [29]. The combination of (4.9) and (4.16) asymptotically guarantees a given level of accuracy, $\mathbf{E}_{TT} < D_1 \text{TOL}_{TT}$. The positive numbers D_1 is motivated to avoid slow convergence in case almost all \bar{r}_n satisfy (4.16), as in Section 4.4.

Now we are ready for the detailed definition of the adaptive algorithm with deterministic steps:

Algorithm D

Initialization Choose:

- (1) an error tolerance, $\text{TOL} \equiv \text{TOL}_S + \text{TOL}_{TT} + \text{TOL}_{TS}$,
- (2) a number, $N[1]$, of initial uniform steps $\Delta \tilde{t}[1]$ for $[0, T]$,
- (3) a number, $M[1]$, of initial realizations and set $M_T[1] = M[1]$,
- (4) a number, $d_1 = 2$ in (4.14) and $c = 1/2 + 1/20$ to compute D_1 using $D_1 > \frac{2}{c}d_1$, and
- (5) a constant $c_0 \geq 1.65$ and an integer $\text{MCH} \geq 2$ to determine the number of realizations in (4.5).

Set the iteration counter, k , for time refinement levels, to 1 and set the statistical error, $\mathbf{E}_{TS} = +\infty$ and $\bar{r}[k] = +\infty$.

Do while ($\bar{r}[k]$ violates the stopping (4.16) or $\mathbf{E}_{TS} > \text{TOL}_{TS}$)

 Compute the sample averages and the error estimates on $\Delta \tilde{t}[k]$
 by calling **Euler**. Set $M_T[k+1] = M_T[k]$ and $\Delta \tilde{t}[k+1] = \Delta \tilde{t}[k]$.

If ($\bar{r}[k]$ violates the stopping (4.16))

 For all time steps $i = 1, \dots, N[k]$, do the refinement process (4.14)
 to update $\Delta \tilde{t}[k+1]$ from $\Delta \tilde{t}[k]$.

elseif ($\mathbf{E}_{TS} > \text{TOL}_{TS}$)

 Update $M_T[k+1]$ by **change_M** ($M_T[k], \mathcal{S}_{TS}[k], \text{TOL}_{TS}; M_T[k+1]$).

end-if

 Increase k by 1.

end-do

Compute an approximation, Eg , for $E[g(\bar{X}(T))]$ with fixed time mesh $\Delta \tilde{t} = \Delta \tilde{t}[k]$

by **Monte-Carlo**($\text{TOL}_S, M_T[k]; Eg$) in (4.4).

Accept Eg as an approximation of $E[g(X(T))]$, since the estimate of the computational error is bounded by TOL .

routine Euler

 For each $M_T[k]$ new realizations, sample jump times with their corresponding marks.

 As described in Section 1.2, use $\Delta \tilde{t}[k]$ to compute corresponding realizations of the Euler method.

 Update the approximations of the time discretization error

 indicators $\bar{r}[k]$ and the statistical time discretization error $\mathbf{E}_{TS}[k]$ and

 compute the sample standard deviation $\mathcal{S}_{TS}[k] \equiv \mathcal{S}(g(\bar{X}(T)); M_T[k])$.

end-of-Euler

4.4. The stochastic time stepping algorithm. Now we describe an adaptive algorithm with stochastic time steps based on a cut-off of the error density $\tilde{\rho}$ introduced in (3.9) of Theorem 3.3, ρ_s , defined by as

$$(4.17) \quad \rho_s^n \equiv \min \left(\max \left(|\tilde{\rho}(t_n, \overline{X})|, \text{TOL}^{\frac{1}{6}} \right), \text{TOL}^{-1} \right), \quad n = 1, \dots, N_A.$$

Following the error expansion in Theorem 3.3, the time discretization error is approximated by

$$(4.18) \quad |\mathcal{E}_T| \lesssim E \left[\sum_{n=1}^{N_A} r_n \right]$$

where the error indicator, r_n , is defined by

$$(4.19) \quad r_n \equiv \rho_s^n (\Delta t_n)^2, \quad n = 1, \dots, N_A.$$

In this case it is optimal, cf. [29], to equidistribute the error contributions among all time steps and all realizations. In other words, the goal of the adaptive algorithm is to construct a time partition Δt of $[0, T]$ for each realization such that

$$(4.20) \quad r_n \leq s_1 \frac{\text{TOL}_T}{E[N_A]}, \quad n = 1, \dots, N_A,$$

where $s_1 = 2$, see Remark 3.1 in [29]. Note that in practice the quantity $E[N_A]$ is not known and we can only estimate it by a sample average $\mathcal{A}(N_A; M)$ from the previous batch of realizations. The statistical error $|E[N_A] - \mathcal{A}(N_A; M)|$ is then bounded by $\mathbf{E}_S(N_A; M)$, with probability close to one, by the same argument as in (4.3).

Let $\overline{N}_A[j] \equiv \mathcal{A}(N_A; M[j])$ be the sample average of the final number of time steps in the j -th batch of $M[j]$ realizations. To achieve (4.20) for each realization, start with an initial partition $\Delta t[1]$ and then specify iteratively a new partition $\Delta t[k+1]$, from $\Delta t[k]$, using the following refinement strategy:

$$(4.21) \quad \begin{aligned} &\textbf{for each realization in the } j\text{-th batch} \\ &\quad \textbf{for each time step } n = 1, \dots, N_A[k] \\ &\quad \quad \textbf{if } r_n[k] \geq s_1 \frac{\text{TOL}_T}{\overline{N}_A[j-1]}, \textbf{ then divide } \Delta t_n[k] \textbf{ into 2 uniform substeps.} \\ &\quad \quad \textbf{else let the new step be the same as the old} \\ &\quad \quad \textbf{endif} \\ &\quad \textbf{endfor.} \\ &\textbf{endfor.} \end{aligned}$$

The refinement strategy (4.21) motivates the following stopping criteria: for each realization of the j -th batch

$$(4.22) \quad \textbf{if } \left(\max_{1 \leq n \leq N_A[k]} r_n[k] < S_1 \frac{\text{TOL}_T}{\overline{N}_A[j-1]} \right) \textbf{ then stop,}$$

where $S_1 > \frac{2}{c} s_1$ with $c \approx \frac{1}{2}$, see [29].

Now we are ready for the detailed definition of the adaptive algorithm with stochastic steps:

Algorithm S

Initialization Choose:

- (1) an error tolerance, $\text{TOL} \equiv \text{TOL}_S + \text{TOL}_T$,
- (2) a number $N[1]$ of initial uniform steps $\Delta t[1]$ for $[0, T]$ and set $\overline{N}_A = N[1]$,
- (3) a number $M[1]$ of initial realizations,
- (4) a number $s_1 = 2$ in (4.21) and $c = \frac{1}{2} + \frac{1}{20}$ to compute S_1 using $S_1 > \frac{2}{c} s_1$, and
- (5) a constant $c_0 \geq 1.65$ and an integer $\text{MCH} \geq 2$ to determine the number of realizations in (4.5).

Set the iteration counter for batches $m = 1$ and the stochastic error $\mathbf{E}_S[m] = +\infty$.

Do while ($\mathbf{E}_S[m] > \text{TOL}_S$)

For realizations $j = 1, \dots, M[m]$

 Set $k = 1$ and $r[k] = +\infty$.

Generate the jump times and their marks $(\tau, Z) = \{(\tau_\ell, Z_\ell)\}_{\ell=1}^{\hat{N}}$.
 Start with the initial partition $\Delta t[k]$ and generate $\Delta W[k]$.
 Compute $g(\overline{X}(T))[J]$ and $N[J]$ by **routine Control-Time-Error**.
end-for
 Compute the sample average $Eg \equiv \mathcal{A}(g(\overline{X}(T)); M[m])$, the sample
 standard deviation $\mathcal{S}[m] \equiv \mathcal{S}(g(\overline{X}(T)); M[m])$ and the a posteriori bound
 for the statistical error $\mathbf{E}_S[m] \equiv \mathbf{E}_S(g(\overline{X}(T)), M[m])$ in (4.3).
if ($\mathbf{E}_S[m] > \text{TOL}_S$)
 Compute $M[m+1]$ by **change_M**($M[m], \mathcal{S}[m], \text{TOL}_S; M[m+1]$), cf.
 (4.5), and update $\overline{N}_A = \mathcal{A}(N_A[J]; M[m])$, where the random variable
 $N_A[J]$ is the final number of time steps on each realization.
end-if
 Increase m by 1.
end-do
 Accept Eg as an approximation of $E[g(X(T))]$, since the estimate of the
 computational error is bounded by TOL .
routine Control-Time-Error($\Delta t[k], \Delta W[k], r[k], (\tau, Z); g(\overline{X}(T))[J], N[J]$)
 Do while ($r[k]$ violates the stopping (4.22))
 Compute the Euler approximation $\overline{X}[k]$ in Section 1.2 and the error indicator
 $r[k]$ in (4.8) using the error density (4.17) on $\Delta t[k]$ with the known
 Wiener increments $\Delta W[k]$.
 If ($r[k]$ violates the stopping (4.22))
 For time steps $i = 1, \dots, N[k]$
 Do the refinement process (4.21) to compute $\Delta t[k+1]$ from $\Delta t[k]$
 and compute $\Delta W[k+1]$ from $\Delta W[k]$ using Brownian bridges.
 end-for
 end-if
 Increase k by 1.
 end-do
 Set the number of the final level $J = k - 1$.
end of Control-Time-Error

5. NUMERICAL EXPERIMENTS

This sections shows numerical results from the implementation of the a posteriori error approximation
 formula presented in Section 2 and of the adaptive algorithms described in Section 4. The programs we
 wrote uses double-precision FORTRAN 77 and is based on the code written for the numerical experiments
 in [34]. For the numerical simulation of the uniform distribution $\mathcal{U}(0, 1)$ and the normal distribution
 $\mathcal{N}(0, 1)$, it applies a double-precision modification of the functions **ran1** and **gasdev** proposed in [32],
 provided an initial seed which must be a negative integer. In particular we use **iseed** for the simulation of
 the Wiener process increments, **zseed** for the simulation of the jump marks and **tseed** for the simulation
 of the jump times.

To perform our computations, we consider a system of stochastic differential equations of the form
 (1.1) with: $d = 2, \ell_0 = \ell_1 = 1$,

$$\begin{aligned}
 a(t, x) &= \left(-x_2, x_1 + \frac{1}{2} \lambda(t) x_2 \right), & b^1(t, x) &= \left(\sqrt{\frac{\lambda(t)}{1+t}} \sin(x_1), 0 \right), \\
 c(t, x, z) &= \left(0, z \frac{\cos(x_1)}{\sqrt{t+1}} - x_2 \right)
 \end{aligned}$$

and a time dependent intensity $\lambda(t) = (1+t)^{-1}$. The distribution for the jump marks is time dependent
 and such that $E[Z_k^2] = 1$. In particular, we use

$$Z_k = \cos(2\pi\tau_k) + \sin(2\pi\tau_k) 2\sqrt{3}(U_k - \frac{1}{2})$$

where $\{\tau_k\}$ are the jumps constructed in Remark 1.1 and $\{U_k\}$ is a sequence a sequence of $\mathcal{U}(0, 1)$ i.i.d. random variables. In this case, due to assumed for of $\lambda(t)$ the inverse function Λ^{-1} is given explicitly by $\Lambda^{-1}(s) = \exp(s) - 1$. This example is a generalization of Example 5.1 in [24], as here we admit a time dependent intensity of the underlying Poisson process and also time dependent distribution of the marks. Taking $g(x) = |x|^2$, $T = 1$ and $X(0) = (0, 0)$ the exact solution of the corresponding weak approximation problem is given by the formula

$$E[g(X(T))] = |X(0)|^2 + \int_0^T \frac{\lambda(s)}{1+s} ds = 1 - \frac{1}{1+T} = \frac{1}{2}.$$

The value of the parameters needed in the simulations are $\text{MCH} = 10$, $c_0 = 1.65$, $\text{iseed} = -7$, $\text{zseed} = -101$ and $\text{tseed} = -20$. We note that the expected value of the number of jumps points equals $\Lambda(T) = \Lambda(1) = \log(2) \simeq 0.693$. Therefore, the computational cost of including the jump times of the process X in our discretization is fairly low, see also the sampled values of $\max \hat{N}$ in Table 5.3, and it is asymptotically negligible as the required accuracy, TOL , tends to zero.

5.1. Deterministic time step algorithm. First, we perform overkilling runs in order to test how realistic is the a posteriori error approximation of the time discretization error described in Theorem 2.3. The results, shown in Table 5.1, show that the ratio of the computational error and its computable approximation tends to 1 as the number of uniform time steps N increases. For each value of N , we choose the number of realizations M large enough in order to keep the total statistical error at the level of 1% of the size of the obtained approximation of the time discretization error.

Alg. D	$c_0 = 1.65$	$\text{iseed} = -7$	$A := \frac{E_T - E_S - E_{TS}}{\mathcal{E}_c}$	$B := \frac{E_T + E_S + E_{TS}}{\mathcal{E}_c}$
N	M	E_T	$E_S + E_{TS}$	$[\min\{A, B\}, \max\{A, B\}]$
5	10×10^6	-0.0602	5.87×10^{-4}	[1.026, 1.046]
10	50×10^6	-0.0314	2.33×10^{-4}	[1.019, 1.035]
20	100×10^6	-0.0159	1.54×10^{-4}	[1.008, 1.028]

TABLE 5.1. Computing the efficient index of the **Algorithm D**.

Table 5.2 contains the results of the **Algorithm D** with an adaptive choice of both the deterministic time steps and the number of realizations. The program starts with $M = 100$ and $N = 5$ subintervals as an initial uniform partition of the time interval $[0, T] = [0, 1]$. The tolerance $\text{TOL} = 0.02$ is divided into $\text{TOL}_S = 0.01333$, $\text{TOL}_T = 0.00444$ and $\text{TOL}_{TS} = 0.00222$. When the algorithm stops, the size of the total approximation error is less than 2TOL according to the stopping criterion (4.16) and it agrees with the size of the computational error.

Alg. D		$c_0 = 1.65$	MCH = 10	iseed=-7	TOL = 0.02	
Iter.	N	M	\mathcal{E}_c	E_T	E_{TS}	E_s
1	5	100	-0.03272	-0.05889	0.02112	0.14602
2	5	1000	-0.04844	-0.06005	0.00684	0.04835
3	5	10000	-0.06592	-0.06070	0.00232	0.01675
4	5	12088	-0.05215	-0.05925	0.00204	0.01459
5	10	12088	-0.03738	-0.03196	0.00107	0.01403
6	20	12088	-0.02585	-0.01633	0.00054	0.01369
7	20	14122	-0.02559	—	—	0.01261

TABLE 5.2. Adaptive choice of M and Δt with Algorithm D.

5.2. Stochastic time step algorithm. To observe the performance of the stochastic time steps **Algorithm S**, we apply it for different values of TOL, starting with a number of realizations $M = 100$ and a number of uniform time steps $N = 5$. Table 5.3 contains the obtained results which show that **Algorithm S** is also effective in giving us an approximation of the quantity of interest within the margin of 2TOL due to the criterion (4.16).

TOL	M	$\mathcal{A}(N_A; M)$	$\min N_A$	$\max N_A$	$\mathcal{S}(N_A; M)$	$\max \hat{N}$	E_s	\mathcal{E}_C
0.040	3.2×10^3	8.3	5	41	5.2	6	2.5×10^{-2}	-2.2×10^{-3}
0.020	12.0×10^3	10.9	5	73	9.2	6	1.3×10^{-2}	-6.0×10^{-3}
0.010	47.9×10^3	20.5	10	146	16.9	6	6.6×10^{-3}	-1.0×10^{-2}
0.005	186.4×10^3	30.1	10	193	32.2	7	3.3×10^{-3}	-3.2×10^{-3}

TABLE 5.3. Adaptive choice of M and Δt with Algorithm S.

Acknowledgements. The work has been supported by: (i) the Swedish Research Council for Engineering Science (TFR) Grant# 222-148, (ii) the Swedish National Network in Applied Mathematics (NTM), (iii) the Project 10.101 FCE - DINACYT, Uruguay, (iv) CSIC - Udelar, Human Resources Program, and (v) the Facultad de Ciencias, Udelar, Montevideo, Uruguay.

APPENDIX A. DISCRETE DUAL EQUATIONS

This appendix section is dedicated to the determination of the discrete dual functions $\varphi(t) \in \mathbb{R}^d$, $\varphi'(t) \in \mathbb{R}^{d \times d}$ and $\varphi''(t) \in \mathbb{R}^{d \times d \times d}$ (see Theorem 2.3 and Theorem 3.3), where t is a node of the (stochastic) partition of the time interval $[0, T]$ which is used by the Euler method (see Section (1.2)). First, introduce the auxiliary functions \hat{A}_i and \hat{c}_i , defined by

$$\begin{aligned}\hat{A}_i(t_n, x) &\equiv x_i + \Delta t_n a_i(t_n, x) + \Delta W_n^\ell b_i^\ell(t_n, x) \quad \forall x \in \mathbb{R}^d, \quad i = 1, \dots, d, \\ \hat{c}_i(t, x, z) &\equiv x_i + c_i(t, x, z) \quad \forall x \in \mathbb{R}^d, \quad \forall z \in \mathbf{Z}, \quad i = 1, \dots, d.\end{aligned}$$

Then, for each realization, φ , φ' and φ'' are constructed by the following algorithm:

Dual backward time stepping algorithm.

Set the initial backward values

$$(A.1) \quad \begin{aligned} \varphi_i(t_{N_A}) &= \partial_i g(\bar{X}(t_{N_A})), \quad \varphi'_{ik}(t_{N_A}) = \partial_{ik} g(\bar{X}(t_{N_A})), \\ \varphi''_{ikm}(t_{N_A}) &= \partial_{ikm} g(\bar{X}(t_{N_A})). \end{aligned}$$

for $n = N_A - 1, \dots, 0$

if $(t_{n+1}$ is a jump time) **then** set $\mathcal{P}^{n+1} = (t_{n+1}^-, \bar{X}(t_{n+1}^-), Z_{n+1})$ and

$$(A.2) \quad \begin{aligned} \varphi_i(t_{n+1}^-) &= \partial_i \hat{c}_j(\mathcal{P}^{n+1}) \varphi_j(t_{n+1}), \\ \varphi'_{ik}(t_{n+1}^-) &= \partial_i \hat{c}_j(\mathcal{P}^{n+1}) \partial_k \hat{c}_p(\mathcal{P}^{n+1}) \varphi'_{jp}(t_{n+1}) + \partial_{ik} \hat{c}_j(\mathcal{P}^{n+1}) \varphi_j(t_{n+1}), \\ \varphi''_{ikm}(t_{n+1}^-) &= \partial_i \hat{c}_j(\mathcal{P}^{n+1}) \partial_k \hat{c}_p(\mathcal{P}^{n+1}) \partial_m \hat{c}_r(\mathcal{P}^{n+1}) \varphi''_{jpr}(t_{n+1}) \\ &\quad + \partial_{im} \hat{c}_j(\mathcal{P}^{n+1}) \partial_k \hat{c}_p(\mathcal{P}^{n+1}) \varphi'_{jp}(t_{n+1}) \\ &\quad + \partial_i \hat{c}_j(\mathcal{P}^{n+1}) \partial_{km} \hat{c}_p(\mathcal{P}^{n+1}) \varphi'_{jp}(t_{n+1}) \\ &\quad + \partial_{ik} \hat{c}_j(\mathcal{P}^{n+1}) \partial_m \hat{c}_p(\mathcal{P}^{n+1}) \varphi'_{jp}(t_{n+1}) \\ &\quad + \partial_{ikm} \hat{c}_j(\mathcal{P}^{n+1}) \varphi_j(t_{n+1}), \end{aligned}$$

else set

$$(A.3) \quad \varphi_i(t_{n+1}^-) = \varphi_i(t_{n+1}), \quad \varphi'_{ij}(t_{n+1}^-) = \varphi'_{ij}(t_{n+1}), \quad \varphi''_{ikm}(t_{n+1}^-) = \varphi''_{ikm}(t_{n+1}).$$

end-if

Set $\hat{\mathcal{P}}^n = (t_n, \bar{X}(t_n))$ and

$$(A.4) \quad \begin{aligned} \varphi_i(t_n) &= \partial_i \hat{A}_j(\hat{\mathcal{P}}^n) \varphi_j(t_{n+1}^-), \\ \varphi'_{ik}(t_n) &= \partial_i \hat{A}_j(\hat{\mathcal{P}}^n) \partial_k \hat{A}_p(\hat{\mathcal{P}}^n) \varphi'_{jp}(t_{n+1}^-) + \partial_{ik} \hat{A}_j(\hat{\mathcal{P}}^n) \varphi_j(t_{n+1}^-), \\ \varphi''_{ikm}(t_n) &= \partial_i \hat{A}_j(\hat{\mathcal{P}}^n) \partial_k \hat{A}_p(\hat{\mathcal{P}}^n) \partial_m \hat{A}_r(\hat{\mathcal{P}}^n) \varphi''_{jpr}(t_{n+1}^-) \\ &\quad + \partial_{im} \hat{A}_j(\hat{\mathcal{P}}^n) \partial_k \hat{A}_p(\hat{\mathcal{P}}^n) \varphi'_{jp}(t_{n+1}^-) \\ &\quad + \partial_i \hat{A}_j(\hat{\mathcal{P}}^n) \partial_{km} \hat{A}_p(\hat{\mathcal{P}}^n) \varphi'_{jp}(t_{n+1}^-) \\ &\quad + \partial_{ik} \hat{A}_j(\hat{\mathcal{P}}^n) \partial_m \hat{A}_p(\hat{\mathcal{P}}^n) \varphi'_{jp}(t_{n+1}^-) + \partial_{ikm} \hat{A}_j(\hat{\mathcal{P}}^n) \varphi_j(t_{n+1}^-). \end{aligned}$$

end-for

Observe that the estimate in Theorem 2.3 needs only the computation of φ and φ' and it is therefore less expensive per realization. Also, with respect to the actual implementation of the dual backward time stepping it is useful to notice that the blocks (A.2) and (A.4) differ only in the function call \hat{c} and \hat{A} .

REFERENCES

- [1] V. M. Alekseev, *An estimate for the perturbations of the solutions of ordinary differential equations. II*, Vestnik Moskov. Univ. Ser. I Mat. Mech **3** (1961), pp. 3–10 (in russian).
- [2] Andersen and Andreasen, *Jump diffusion models: volatility smile fitting and numerical methods for pricing*. Review of derivatives research. **4** (2000), pp. 231–262.
- [3] I. Babuška, A. Miller and M. Vogelius, *Adaptive methods and error estimation for elliptic problems of structural mechanics* in Adaptive computational methods for partial differential equations (College Park, Md., 1983), SIAM, Philadelphia, Pa., (1983), pp. 57–73.
- [4] V. Bally and D. Talay, *The Euler scheme for stochastic differential equations: error analysis with Malliavin calculus*. Math. Comput. Simulation **38** (1995), pp. 35–41.
- [5] V. Bally and D. Talay, *The law of the Euler scheme for stochastic differential equations. II. Convergence rate of the density* Monte Carlo Methods Appl. **2** (1996), pp. 93–128.
- [6] R. Becker and R. Rannacher, *A feed-back approach to error control in finite element methods: basic analysis and examples*. East-West J. Numer. Math., **4** (1996), pp. 237–264.
- [7] N. Bruti-Liberati and E. Platen, *Weak Numerical Methods for Jump-Diffusion Processes with Applications in Finance* Presentation in the Conference: “Numerical Methods in Finance (Inria-Rocquencourt, France, 1/2/2006-3/2/2006)”
- [8] R. Cont and P. Tankov, *Financial Modelling with Jump Processes*, Chapman and Hall, Financial Mathematics Series, (2004).

- [9] R. Cont and P. Tankov, *Calibration of jump-diffusion option pricing models: a robust non-parametric approach*, working paper.
- [10] R. Durrett, *Probability: theory and examples*, Duxbury Press, Belmont CA, 1964.
- [11] A. Dzougoutov, K.-S. Moon, E. von Schwerin, A. Szepessy, R. Tempone. *Adaptive Monte Carlo algorithms for stopped diffusion* in Multiscale methods in science and engineering, pp. 59–88, Lect. Notes Comput. Sci. Eng., vol. 44, Springer, Berlin, 2005.
- [12] K. Eriksson, D. Estep, P. Hansbo and C. Johnson, *Introduction to adaptive methods for differential equations*, Acta Numerica, (1995), 105–158.
- [13] G. S. Fishman, *Monte Carlo: Concepts, Algorithms, Applications*, Springer-Verlag, New York, 1996.
- [14] A. Friedman, *Partial differential equations of parabolic type* Prentice-Hall, Englewood Cliffs, New Jersey (1964)
- [15] P. Glasserman and N. Merener. *Numerical Solution of Jump-Diffusion LIBOR Market Models*. Finance Stoch. **7** (2003), no. 1, 1–27.
- [16] D. J. Higham and P. E. Kloeden, *Numerical methods for nonlinear stochastic differential equations with jumps*. Numer. Math. **101** (2005), no. 1, 101–119;
- [17] N. Ikeda and S. Watanabe, *Stochastic differential equations and diffusion processes*, Second edition, North-Holland, Amsterdam, (1989)
- [18] Jacod, J. and Shiryaev, A.N., *Limit Theorems for Stochastic Processes*. Springer, Berlin, Heidelberg. (1987)
- [19] C. Johnson, *Error estimates and adaptive time-step control for a class of one-step methods for stiff ordinary differential equations* SIAM J. Numer. Anal. **25** (1988), pp. 908–926
- [20] C. Johnson and A. Szepessy. *Adaptive finite element methods for conservation laws based on a posteriori error estimates*. Comm. Pure Appl. Math. **48** (1995), pp. 199–234.
- [21] P.E. Kloeden and E. Platen, *Numerical solution of stochastic differential equations*, Applications of Mathematics, **23**, Springer-Verlag, New York, 1992.
- [22] I. Karatzas and S.E. Shreve, *Brownian motion and stochastic calculus*, Graduate Texts in Mathematics 113, Springer-Verlag, New York, 1988.
- [23] A. L. Lewis. *A simple option formula for general jump-diffusion and other exponential Lévy processes*. Envision Financial Systems and OptionCity.net. September 2001.
- [24] X. Q. Liu and C. W. Li. *Weak approximation and extrapolations of stochastic differential equations with jumps*. SIAM J. Numer. Anal. **37** (2000), pp. 1747–1767.
- [25] A.-M. Matache, T. von Petersdorff and C. Schwab, *Fast deterministic pricing of options on Lévy driven assets*, Report 2002-11.
- [26] G. N. Milstein, *Numerical integration of stochastic differential equations*, Mathematics and its Applications, **313**, Kluwer Academic Publishers, Dordrecht, 1995.
- [27] K.-S. Moon, E. von Schwerin, A. Szepessy and R. Tempone, *An adaptive algorithm for Ordinary, Stochastic and Partial Differential Equations*, Contemporary Mathematics **383** (2005), pp. 325–343.
- [28] K.-S. Moon, A. Szepessy, R. Tempone and G.E. Zouraris, *Hyperbolic differential equations and adaptive numerics*, in Theory and Numerics of Differential Equations (Durham, 2000) Eds. Blowey, J. F.; Coleman, J. P.; Craig, A. W. pp. 231–280, Universitext, Springer, Berlin, 2001.
- [29] K.-S. Moon, A. Szepessy, R. Tempone and G.E. Zouraris, *Convergence rates for adaptive weak approximation of stochastic differential equations*, Stochastic Analysis and Applications **23** (2005), pp. 511–558.
- [30] D. Nualart, *The Malliavin calculus and related topics*, Probability and its Applications, Springer-Verlag, New York, 1995.
- [31] B. Øksendal, *Stochastic differential equations*, Fifth edition, Springer-Verlag, Berlin, 1998.
- [32] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical recipes in FORTRAN. The art of scientific computing*, Second edition, Cambridge University Press, Cambridge, 1992.
- [33] P. Protter, *Stochastic integration and differential equations*, Applications of Mathematics 21, Springer-Verlag, Berlin, 1990.
- [34] A. Szepessy, R. Tempone and G.E. Zouraris. *Adaptive weak approximation of stochastic differential equations*. Comm. Pure Appl. Math. **54** (2001), pp. 1169–1214.
- [35] D. W. Strook, *Lectures on topics in stochastic differential equations*, TATA Institute of Fundamental Research, Bombay, India, 1982.
- [36] D.W. Strook and S.R.S. Varadhan. *Multidimensional diffusion processes*. Grundlehren der mathematischen Wissenschaften 233, Springer-Verlag, Berlin-New York, 1979.
- [37] D.W. Strook and S.R.S. Varadhan, *Diffusion processes with continuous coefficients, I and II* Comm. Pure Appl. Math. **22** (1969), pp. 345–400 and 479–530.
- [38] P. Protter and D. Talay. *The Euler scheme for Lévy driven stochastic differential equations*. Ann. Probab. **25** (1997), pp. 393–423.
- [39] D. Talay and L. Tubaro. *Expansion of the global error for numerical schemes solving stochastic differential equations*. Stochastic Anal. Appl. **8** (1990), pp. 483–509.